NETWORK INTERDICTION APPROACHES FOR DIMINISHING MISINFORMATION SPREAD IN SOCIAL NETWORKS

By

NILOUFAR DAEMI

Bachelor of Science in Industrial Engineering
Noshirvani University of Technology
Babol, Iran
2012

Master of Science in Industrial Engineering
Amirkabir University of Technology
Tehran, Iran
2016

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 2022

NETWORK INTERDICTION APPROACHES FOR DIMINISHING MISINFORMATION SPREAD IN SOCIAL NETWORKS

Dissertation Approved:

Dr. Juan S. Borrero

Dissertation Advisor

Dr. Balabhaskar Balasundaram

Dr. Austin Buchanan

Dr. Charles Chen

ACKNOWLEDGMENTS

Dedicated to

Mahsa Amini,

People of Sistan and Baluchestan,

and all the brave Iranians who are fighting for freedom.

Acknowledgments reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Juan Borrero, for his continuous support and encouragement during my Ph.D. I really appreciate the immense respect and support he has always shown me in these years that helped me get through the most challenging moments of my Ph.D. life. None of the achievements during my Ph.D. could have been completed without his guidance.

I am also very grateful to Dr. Balabhaskar Balasundaram for his support throughout these years. I can never thank him enough for all he has taught me. His impact was not limited to his expertise in the field; I have been also much influenced by his genuine personality to the extent that he is a role model for me.

I would like to thank my committee members, Dr. Balabhaskar Balasundaram, Dr. Austin Buchanan, and Dr. Charles Chen for their guidance and insightful comments. Additionally, I thank Drs. Austin Buchanan, Hosseinali Salemi, and Hamidreza Validi for helping me integrate their work with part of my research. Their assistance has been extremely valuable to me.

Finally, I am thankful to my parents for their love, and support, and the sacrifices they made so I can follow my dreams. When I shared with them my decision to begin a Ph.D. in the US, they showed nothing but encouragement although this decision was going to separate us for years. They have always been a reason for me to make every effort to grow and every day away from them, I have lived with the dream of becoming the girl they can be proud of.

Acknowledgments reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: NILOUFAR DAEMI

Date of Degree: DECEMBER, 2022

Title of Study: NETWORK INTERDICTION APPROACHES FOR DIMINISHING MISINFORMATION SPREAD IN SOCIAL NETWORKS

Major Field: INDUSTRIAL ENGINEERING AND MANAGEMENT

Abstract: Network interdiction has many applications in many domains, including telecommunications, epidemic control, and social network analysis. In this dissertation, we use network interdiction to devise strategies for the problem of misinformation dissemination in online social networks. These platforms provide the opportunity of quick communication between users, which in a network with malicious accounts can result in the fast spread of rumors and harmful content. We study this topic based on two different approaches. The first approach focuses on interdicting cohesive subgroups of malicious accounts. We use s-clubs, which are subsets of vertices that induce subgraphs of diameter at most s to model the cohesive social subgroups. We consider a defender that can disrupt the vertices of the adversarial network to minimize its threat, which leads us to consider a maximum s-club interdiction problem. Using a new notion of H-heredity in s-clubs, we provide a mixed-integer linear programming formulation for this problem that uses far fewer constraints than the formulation based on standard techniques. We further relate H-heredity to latency-s connected dominating sets and design a decomposition branch-and-cut algorithm for the problem. The second methodology that is studied in this dissertation is to delay the spread of misinformation in the network using first passage times interdiction. The first passage times are defined as the first time each user is exposed to a post shared by another user in the network and is computed using a discrete time Markov chain model. Vertices are interdicted to modify the transition probabilities and increase the propagation times between users who share misinformation and harmful content, and vulnerable users. We show that the problem is NP-hard and provide a mixed-integer linear programming formulation for it. Computational experiments on benchmark instances are conducted for both interdiction approaches based on cohesive subgroups and first passage times in order to assess the computational capabilities of the methods we introduced.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure | Page

CHAPTER I

INTRODUCTION

Online social networks are constantly developing and have changed the way people communicate. Despite their benefits, online social networks allow the spread of misinformation and rumors due to their openness and lack of fact-checking tools which can result in economic losses (Domm, 2013) or negative impacts on the public (Allcott and Gentzkow, 2017). There is empirical evidence that bots and malicious accounts play a critical role to increase anxiety or affect the stock market and elections. One example is the tweets containing fake images of hurricane Sandy in 2012 that led to panic and chaos among people (Gupta et al., 2013). Another example is wiping out $130 billion in stock value in 2013, immediately after a false tweet claiming that Barack Obama was injured in an explosion (Rapoza, 2017). Regarding the effect of the malicious accounts' activities on the elections, analysis shows that during the three months before the US presidential election in 2016, fake news on Facebook generated more engagement in comparison to real news from prominent news outlets (Silverman, 2016).

In addition to the issues caused by fake news, some content on social networks can be harmful to vulnerable users including people of young or old age, and people with mental health challenges. Studies show a correlation between the use of social media and depression in teenagers and surveys reveal that teenagers report direct negative impacts of social media on their happiness and self-esteem (Cochet, 2021; Youth Equipped To Succeed, 2022). This motivates the development of strategies to prevent the spread of misinformation and harmful

Parts of this document are reprinted with permission from Daemi et al. (2022)

content on these online platforms.

The problem of misinformation blocking in networks has been widely studied in the literature using different strategies. Minimizing the spread of misinformation with a limited budget, setting a threshold on the influence of misinformation, and finding the misinformation source in the network are some examples of different approaches to this problem. Most of the available studies formulated graph models and proposed greedy and heuristic algorithms to solve them (Tanınmış et al., 2020; Pham et al., 2019; Shi et al., 2019; Shah and Zaman, 2016).

In this dissertation, we study misinformation blocking in networks using two different network interdiction models and develop integer programming techniques to solve them. Unlike several existing studies with heuristic approaches, our methodologies provide optimal solutions to minimize the spread of misinformation. In addition, instead of focusing on detecting the rumor source or malicious bots, our models provide strategies to interdict users based on their ability to spread misinformation by considering the structure of the network, e.g., membership of a user in cohesive subgroups. This approach is justified based on the fact that even if rumors are initiated by malicious accounts, innocent users might be unwittingly helping with the spread of rumors, and considering them in strategies can be beneficial.

The first approach introduced in this dissertation is to model this problem based on the concept of cohesive subgroups in social networks. Cohesive subgroups can represent groups of individuals in social networks that share core beliefs, influence each other, and act as a unit together towards a common goal (Wasserman and Faust, 1994). When such cohesive subgroups contain malicious accounts, they are able to effectively propagate undesirable rumors in the network (Help Net Security, 2019). Therefore, it is reasonable to consider minimizing the size of these cohesive subgroups of adversarial accounts (by temporarily disabling the accounts) as an approach to prevent the spread of misinformation in networks.

For this problem, we propose a methodology to minimize the size of the largest cohesive subgroups which may contain malicious accounts. We define an optimization problem to find the best interdiction policies assuming there is an interdiction penalty to delete vertices. In this setting, deleting vertices is equivalent to disabling the users in the social network. In Chapter III, we discuss this methodology in detail.

The second approach is to delay the propagation of misinformation through the network. In this setting, instead of disabling the users, the focus is on reducing the speed of misinformation dissemination. We use the concept of the first passage time in a Markov chain. Based on this concept, we define the first passage time between user i and user j in a social network as the first time user j is exposed to a post that has been shared by user i. We use the first passage time as a measure that shows how fast the misinformation spreads in the network. The goal is to optimally increase the value of the expected first passage time from a group of users who spread harmful content denoted by S to the group of vulnerable users denoted by T.

We model this problem as the maximization of the smallest first passage time from vertices in S to vertices in T assuming there is a limited budget to interdict vertices. In this setting, a transition probability matrix is used which describes how a message moves throughout the users of the network. Interdicting a vertex decreases the probability that the user successfully shares a post with their neighbors in the network. In Chapter V, we discuss this approach in detail.

1.1 Graph notations and definitions

Consider a graph $G = (V, E)$ with vertex set $V \coloneqq \{1, \ldots, n\}$ and edge set $E \subseteq \binom{V}{2} \coloneqq \{\{u, v\} \mid u, v \in V, v \neq u\}$. We will assume throughout that G is not an empty graph, i.e., $E \neq \emptyset$. Denote by $N_G(v) \coloneqq \{u \in V \mid \{u, v\} \in E\}$, the set of neighbors of vertex v and its cardinality by $\deg_G(v)$. We also use the notation $N_G[v] \coloneqq N_G(v) \cup \{v\}$ to denote the *closed* neighborhood of a vertex v in G. We denote the subgraph induced by a set of vertices $S \subseteq V$ by $G[S] \coloneqq \left(S, \binom{S}{2} \cap E\right)$. For convenience, we denote the deletion of a set of vertices T and incident edges as $G \setminus T \coloneqq G[V \setminus T]$.

Let $\text{dist}_G(u, v)$ denote the length of a shortest path between a connected pair for vertices u and v in G, where the length of a path is the number of edges in the path. The diameter of a connected graph is the maximum distance between a pair of vertices, and we denote it by $\text{diam}(G) \coloneqq \max\{\text{dist}_G(u, v) \mid u, v \in V\}$. If u and v are in different connected components of G, then the distance between them, and hence the diameter of that disconnected graph are taken to be infinite. Also, we assume that $\text{dist}_G(u, u) = 0$. When the graph G under consideration is known without any ambiguity, we drop the subscript G for convenience from all the notations. Below, we present the formal definition of structures and concepts that will be used in Chapters III–V. Figure 1.1 illustrates the structures and properties introduced in Definitions 1–3.

Definition 1 (Luce and Perry (1949))**.** *A subset of vertices $C \subseteq V$ is called a clique if the induced graph $G[C]$ is complete, i.e., all the vertices in $G[C]$ are adjacent.*

Definition 2 (Mokken (1979))**.** *Given a graph $G = (V, E)$ and a positive integer s, we call a subset of vertices $S \subseteq V$ an s-club if* $\text{diam}(G[S]) \leq s$.

The largest cardinality of an s-club is called *the s-club number* of graph G, denoted by $\bar{\omega}_s(G)$. Detecting a maximum cardinality s-club, i.e., *the maximum s-club problem*, is NP-hard in general (Bourjolly et al., 2002) and in graphs of diameter $s + 1$ (Balasundaram et al., 2005). The model is one of several types of clique relaxations that have been studied in the literature (Pattillo et al., 2013), and it reduces to a clique when $s = 1$. With $s = 2$, we obtain a formalization of the friend-of-a-friend cluster, as a 2-club S must satisfy at least one of the following conditions for every distinct pair of vertices $u, v \in S$: either $\{u, v\} \in E$, or $N_G(u) \cap N_G(v) \cap S \neq \emptyset$. In other words, every pair of members of a 2-club are either friends or they have a mutual friend in the group. In general, s-clubs for low values of parameter $s \in \{2, 3\}$ can be used to represent clusters where quick communication between members is possible.

Definition 3 (Lewis and Yannakakis (1980)). *Suppose for a vertex subset $S \subseteq V$, the induced graph $G[S]$ satisfies property Π. The graph property Π is said to be hereditary if, by deletion of any subset of vertices of S, the induced graph has property Π.*

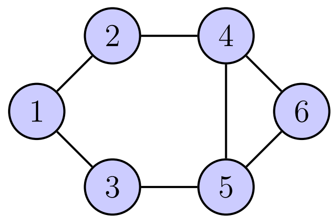

Figure 1.1: In this graph, $C = \{4, 5, 6\}$ forms a clique and by removing any subset $D \subseteq C$, set $C \setminus D$ is still a clique. Set $S = \{1, 2, 3, 4, 5\}$ forms a 2-club, but not every subset of S induces a 2-club.

Definition 4 (Kulkarni (2016)). *A stochastic process $\{X_n, n \geq 0\}$ with countable state-space S is called a Discrete-Time Markov Chain (DTMC), if for all $n \geq 0$, we have $X_n \in S$, and for all $n \geq 0$ and $i, j \in S$: $P(X_{n+1} = j \mid X_n = i, X_{n-1}, X_{n-2}, ..., X_0) = P(X_{n+1} = j \mid X_n = i)$.*

The equation in Definition 4 states the Markov property in DTMC which is that the future state of the system only depends on the present state.

Definition 5 (Kulkarni (2016)). *In a stochastic system, the first passage time from state i to state j denoted by t_{ij} is defined as the first time that the chain hits state j starting from state i. In other words, $t_{ij} = \inf\{n \geq 0 | X_n = j, X_0 = i\}$*

The first passage time in a DTMC is calculated using a transition probability matrix P by solving a system of linear equations with n^2 variables and n^2 equations where n is the number of states in the system:

$$t_{ij} = 1 + \sum_{\substack{k=1 \\ k \neq j}}^{n} P_{ik} t_{kj} + P_{ii} t_{ij} \tag{1.1}$$

where P_{ik} is the probability of transition from state i to state k in a single time unit (Kulkarni, 2016).

CHAPTER II

LITERATURE REVIEW

Network interdiction problems involve an interdictor and an evader where the evader operates a network in order to optimize some objective function and the interdictor changes the structure of the network by deleting vertices or edges to limit the evader's achievable objective value (Israeli and Wood, 2002; Borrero et al., 2016). In this work, we use network interdiction to limit the dissemination of misinformation in online social networks.

In Section 2.1, we review studies on network interdiction problems that are the most relevant to our problem. Also, because we solve the maximum s-club interdiction problem in Chapter III, we will review works on the maximum s-club problem in Section 2.2

2.1 Network interdiction

Network interdiction was initially studied in the context of military applications, but it can be applied to solve problems such as crime detection, prevention of outbreaks of infectious diseases, and online social networks analysis (Sullivan et al., 2014; Assimakopoulos, 1987; Furini et al., 2019).

Shortest path interdiction is an example of an interdiction problem with many applications in the real world. Here, the evader's goal is to move a supply convoy through the network as quickly as possible. The interdictor removes arcs to maximize the shortest path length from an origin to a destination to limit the evader's achievable objective value (Corley and David, 1982; Malik et al., 1989; Israeli and Wood, 2002). The problem is NP-hard (Ball et al.,

1989) and is formulated as a bilevel, max-min optimization problem. However, it can be reformulated as a single-level problem and solved effectively using mixed inter programming techniques (Israeli and Wood, 2002). Models focusing on minimizing the maximum flow by interdiction using a limited budget are solved in a similar manner (Wood, 1993). A comprehensive review of classical interdiction models can be found in Smith and Song (2020). The interdictor in a network interdiction problem might have incomplete information about the network. Borrero et al. (2016) study the shortest path interdiction problem in the setting where the interdictor removes k arcs at each iteration and, based on the evader's solution (the shortest path in the interdicted network), the interdictor obtains information and modifies its interdiction policy. The findings in this study are later generalized to other interdiction and bilevel settings (Borrero et al., 2019, 2022)

Another application of interdiction models is to reduce the spread of infections in networks in cases of disease outbreak or cyber-security threats. Nandi and Medal (2016) study an interdiction problem in a network with infected and susceptible subsets of nodes. The spread of infection is represented by the average number of new infections and the average time to infect half of the susceptible nodes. The interdictor removes edges to minimize different quantities such as the number of connections between infected and susceptible nodes.

The influence minimization problem in networks is also studied where the interdictor protects or deactivates a subset of nodes in the network and the evader uses unprotected nodes to initiate a diffusion process to maximize the total number of influenced nodes. These problems are solved using heuristics and approximation algorithms (Tanınmış et al., 2019, 2020). One of the methods to solve these bilevel problems is the x-space algorithm proposed by Tang et al. (2016) which solves the upper and lower bound problems consecutively until convergence. Recently, an improved version of this algorithm has been proposed by Tanınmış et al. (2022).

Network interdiction techniques can also be applied to reduce the size of cohesive subgroups in a network. Furini et al. (2019) study the maximum clique interdiction problem where, given a budget b, the goal is to find a subset of at most b vertices to remove from the graph so that the clique number in the remaining graph is minimized. They exploit the hereditary property of cliques to formulate the problem and solve several instances including large-scale social networks to optimality. Although this approach can be useful to analyze social networks, the clique structure can be too restrictive as it requires the diameter of the induced subgraph to be one. To address this issue, different clique relaxations such as k-plex (Balasundaram et al., 2011), k-core (Seidman, 1983), and s-club (Mokken, 1979) have been introduced that can be more useful in practice. In Chapter III, we will study the maximum s-club and maximum k-core interdiction problems.

In all of these examples of network interdiction problems, it is assumed that evaders are deterministic and their behavior only depends on their objective such as finding a shortest path in a graph. We can also consider evaders whose actions are described by Markov processes because they have incomplete information about the network or limited planning time.

Gutfraind et al. (2009) study an interdiction model with multiple Markovian evaders who choose edges to traverse based on a random walk defined by a Markovian transition matrix. Each evader has a target in the network, and the goal of the interdictor is to interdict edges using a limited budget to increase the probability of capturing evaders before reaching their targets. They show the NP-hardness of the problem for multiple evaders by a reduction from the set cover problem and provide a greedy algorithm to find the interdiction set.

Johnson et al. (2014) study two interdiction problems with Markovian evaders. One problem is called the budgeted interdiction problem and maximizes the number of captured evaders under a limited budget for node interdiction. The other problem is called the full interdiction problem and its goal is to capture all evaders at minimum cost. They examine the complexity of these problems for different classes of special graphs. The budgeted interdiction version is shown to be NP-hard even with a single evader, while the full interdiction problem with one evader is solvable in polynomial time.

Sefair et al. (2017) consider a system based on a Discrete-Time Markov Chain (DTMC) where the interdictor protects a subset of nodes with a limited budget while the evader attacks a set of unprotected nodes leading to changes in the transition probabilities. The evader's goal is to minimize the Weighted Expected Hitting Time, while the interdictor seeks to maximize it. The authors propose a mixed integer linear programming formulation and a first-order approximation method to solve this problem.

The interdiction of DTMCs has applications in a wide range of systems such as revenue management in air cargo where the used capacity (weight and volume) is modeled as a DTMC and the goal is to maximize an expected revenue function that depends on the DTMC's probabilities (Han et al., 2010) or controlling infectious diseases where a DTMC describes the disease spread (Meltzer et al., 2001). In Chapter V, we study an interdiction problem with a similar setting where, at each step, a transition probability matrix determines the next edge to be traversed to spread misinformation through the network.

2.2 The maximum s-club problem

A subset of vertices in a graph is an s-club if the distance between every pair of vertices is at most s in the induced subgraph (Bourjolly et al., 2000). The largest cardinality of an s-club is called *the s-club number* of graph G, denoted by $\bar{\omega}_s(G)$. Detecting a maximum cardinality s-club, i.e., *the maximum s-club problem*, is NP-hard in general (Bourjolly et al., 2002) and in graphs of diameter $s+1$ (Balasundaram et al., 2005). This model is one of several types of clique relaxations that have been studied in the literature (Pattillo et al., 2013), and it reduces to a clique when $s = 1$.

The maximum s-club problem has been extensively studied. The first IP formulation to solve this problem was proposed by Bourjolly et al. (2002) where a binary variable y_t is defined for every chain t with length at most s that links vertices $i, j \in V$ and a constraint is introduced for every pair of vertices $\{i, j\} \notin E$ which requires at least one of the variables y_t to take a value of one if both vertices i and j are present in the s-club. When $s = 2$, the formulation can be simplified by considering only common neighbors between vertices i and j and defining the constraint $\sum_{r \in N(i) \cap N(j)} x_r \geq x_i + x_j - 1$ where $N(i)$ is the set of neighbors of vertex i.

Lu et al. (2018) proposed an algorithm that exploits the s-clique formulation as a relaxation of the maximum s-club problem. The algorithm is designed based on the fact that every s-club is also an s-clique where an s-clique S is a subset of vertices such that $\text{dist}_G(u, v) \leq s$ for every $u, v \in S$. They proposed a decomposition and branch-and-cut algorithm to eliminate every s-clique in the graph that is not an s-club. They combined this algorithm with preprocessing techniques to determine vertices that satisfy the required condition to be in an s-club based on the size of their distance-s neighborhood. This technique reduces the size of the input graph for which the s-club problem should be solved.

Salemi and Buchanan (2020) define the concepts of length-s a, b-connector and length-s a, b-separator and based on these concepts, they propose two integer programming formulations path-like and cut-like to find a maximum s-club in a graph. Their computational results show that the cut-like formulation outperforms several previous formulations. As a preprocessing step to implement this formulation, they modify the heuristic algorithm presented by Bourjolly et al. (2000) and defined s-th power graph G^s to find a large clique in the graph. As a result, the size of the input graph reduces significantly in a reasonable time. We will use this approach to solve the subproblem in our bilevel model in Chapter III.

CHAPTER III

COHESIVE SUBGROUP INTERDICTION

3.1 Motivation

Cohesive subgroups in social networks can represent groups of individuals that share core beliefs, influence each other, and act together as a unit towards a common goal (Wasserman and Faust, 1994). In more general networks, cohesive subgroup models provide formalizations of "tightly-knit clusters" (Balasundaram et al., 2011) and therefore have been used in applications beyond social network analysis, for example, to analyze complex biological networks (Pasupuleti, 2008; Butenko and Wilhelm, 2006; Balasundaram et al., 2005).

The canonical optimization problem of identifying a particular type of clique relaxation of maximum cardinality (or weight) has received considerable attention in the literature (Pattillo et al., 2013; Balasundaram and Pajouh, 2013). In this chapter, we focus on interdicting a clique relaxation called s-club that models low-diameter clusters. Also, in Section 3.7, we briefly study the maximum k-core interdiction problem.

As a motivating example for the problem of interdicting low-diameter clusters, consider the following stylized scenario. Suppose a social media network manager (NM) recognizes that disinformation is being spread with hashtags `#badrumor` and `#fakenews` and suspects that a coordinated group of adversarial actors whose identities are unknown may be responsible. Although the NM could ban or deactivate accounts, it would not be effective to do so arbitrarily. The NM can consider the following graph model to capture this situation, let us refer to it as the *rumor graph*: the vertex set would include all user accounts using one of the offending hashtags in their posts; the edge set would include an edge $\{u, v\}$ if account u liked or reshared a post by account v that included one of the offending hashtags. We use an undirected edge to indicate that the accounts represented by the end-points are related, and not necessarily that one is *directing* the other. Under the assumption that the interaction patterns of such suspicious accounts in the rumor graph resembles a cohesive social subgroup that is capable of quick communication, one could arguably phrase the NM's decision problem as one of optimally interdicting (by disabling accounts) all large low-diameter cohesive subgroups in the rumor graph.

Although we are describing a stylized version of the decision problem faced by the NM, it can be a reasonable first step in analyzing such problems to devise effective interdiction policies in practice. To begin with, we choose to model cohesive subgroups of interest in this rumor network as *s-clubs* for low values of parameter s that ensure short pairwise distances inside the cohesive subgroup between members as a surrogate for quick communication between group members. We also assume that one of the maximum cardinality s-clubs contains the adversarial accounts and that diminishing its size can impact that group's ability to spread disinformation. Furthermore, the other maximum cardinality s-clubs (those not containing the adversarial actors) are unwittingly helping with the spread of the rumor and arguably also warrant deactivation.

Interdiction by vertex deletion is the focus of this study. Suppose $T \subset V$ is the "deletion set." A fundamental difference between interdicting cliques in a graph (Furini et al., 2019) versus s-clubs in a graph is *heredity.* If $K \subset V$ is a clique in G then $K \setminus T$ is a clique in $G \setminus T$ because the clique property is preserved under vertex deletion. However, the s-club property is not hereditary under vertex deletion; see Figure 3.1 (Alba, 1973). Consequently, if S is an s-club in G, we cannot claim that $S \setminus T$ is an s-club in $G \setminus T$ for every $T \subseteq S$. This fundamental difference drives all of the approaches taken in this work to model and solve the

s-club interdiction problem, and differentiates it from the techniques recently proposed for interdicting cliques (Furini et al., 2019).

Typically, interdiction comes "at a cost." If there were no restrictions on T, the entire graph can be deleted. Shortest path and other network flow interdiction problems are often motivated by applications that justify using a budget b in a constraint that says the size of T cannot exceed b (Morton et al., 2007; Pan et al., 2003; Israeli and Wood, 2002). The budget in these settings is derived from physical restrictions such as the number of patrol vehicles available to intercept smugglers or the number of sensors that can be deployed in the network for monitoring purposes. In our setting, we avoid the use of a hard budget constraint as the NM can delete any number of vertices (e.g., by banning or temporarily disabling user accounts) and may be willing to delete a large number of accounts to stem the rumor (Spangler, 2018). However, if there is no cost incurred by deleting vertices, we set up a pointless and trivial problem that would suggest deleting V. Our focus is on identifying and deleting "club-critical vertices", and we assume that we incur an interdiction penalty in doing so, as opposed to a hard budget constraint.

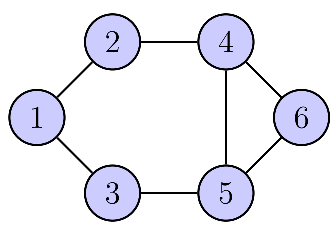

Figure 3.1: The set $S = \{1, 2, 3, 4, 5\}$ is a 2-club. After deleting any vertex $i \in S$, the set $S \setminus \{i\}$ will not be a 2-club.

3.2 Problem statement

We wish to solve the following optimization problem to find an *optimal interdiction policy*, that is, a subset of vertices T^* that achieves the following minimum:

$$\min_{T \subseteq V} \left\{ \bar{\omega}_s(G \setminus T) + \alpha|T| \right\}, \tag{3.1}$$

where $\alpha > 0$ is the unit penalty cost of deleting a vertex. We could interpret this choice of penalty as follows. As the empty set is a feasible solution to problem (3.1), we have:

$$\begin{aligned} & \bar{\omega}_s(G \setminus T^*) + \alpha|T^*| \leq \bar{\omega}_s(G) \\ \implies & \frac{\bar{\omega}_s(G) - \bar{\omega}_s(G \setminus T^*)}{|T^*|} \geq \alpha, \text{ assuming } T^* \neq \emptyset. \end{aligned}$$

The ratio of the decrease in the s-club number upon interdiction to the size of an optimal deletion set (when non-empty) is at least α. In our models, we typically choose $\alpha \in \bigcup_{k \in \mathbb{N}} \{k, \frac{1}{k}\}$. By setting $\alpha = k$, the NM can use an operating policy that requires the s-club number decreases by at least k for each vertex deleted. In settings where we are prepared to delete a large number of vertices to decrease the s-club number, we can delete up to k times the decrease that we can produce by setting $\alpha = 1/k$.

3.3 A preliminary formulation

An MILP formulation of problem (3.1) can be derived by using standard techniques in interdiction (Fischetti et al., 2018, 2019; Smith and Song, 2020). To this end, we use vectors $x \in \{0,1\}^{|V|}$ as incidence vectors of a deletion set, thus $x_v = 1$ if v is deleted and zero otherwise. We let T^x denote the set of vertices deleted in solution x, thus $T^x = \{v \in V \mid x_v = 1\}$. Henceforth, we also use the convenient short form $x(S)$ in place of $\sum_{v \in S} x_v$ for $S \subseteq V$. In terms of x, problem (3.1) is given by:

$$z_{s,\alpha} = \min \left\{ \bar{\omega}_s(G \setminus T^x) + \alpha x(V) \;\middle|\; x \in \{0,1\}^{|V|} \right\}. \tag{3.2}$$

The bilevel optimization problem (3.2) can be reformulated as the following single-level MILP:

$$\begin{aligned} z_{s,\alpha} = \min\ & \theta + \alpha x(V) && (3.3a)\\ s.t.\quad & \theta \geq |S| - |S|x(S) \qquad \forall S \in \mathcal{S} && (3.3b)\\ & x \in \{0,1\}^{|V|}, \theta \in \mathbb{R}_+, && (3.3c) \end{aligned}$$

where $\mathcal{S}$ is the collection of all s-clubs in G. The right-hand side of Constraint (3.3b) becomes redundant if a vertex in S is interdicted. Otherwise, the cardinality of the maximum s-club in the interdicted graph $G \setminus T^x$ should be at least $|S|$.

Although valid, the direct implementation of Formulation (3.3) in an MILP solver is untenable for large instances as it requires enumerating exponentially many s-clubs in G in the worst case. Nevertheless, this formulation can be used in a delayed constraint generation algorithm as follows. Let $\mathcal{S}^0 \subseteq \mathcal{S}$ be an initial collection of s-clubs. In iteration $i = 0, 1, \ldots$, the algorithm solves the initial relaxation problem:

$$\min \left\{ \theta + \alpha x(V) \;\middle|\; \theta \geq |S| - |S|x(S)\ \forall S \in \mathcal{S}^i, x \in \{0,1\}^{|V|}, \theta \in \mathbb{R}_+ \right\} \tag{3.4}$$

and recovers an optimal solution (θ^i, x^i). If $\theta^i \geq \bar{\omega}_s(G \setminus T^{x^i})$ then it follows that (θ^i, x^i) is an optimal solution to problem (3.3), and the algorithm terminates. Otherwise, the algorithm identifies an s-club S' in $G \setminus T^{x^i}$ such that $|S'| > \theta^i$ and updates $\mathcal{S}^{i+1} := \mathcal{S}^i \cup \{S'\}$.

Clearly, this delayed constraint generation algorithm converges to an optimal solution of problem (3.3) in a finite number of steps because $\mathcal{S}$ is a finite set. However, it has two important limitations. First, each iteration requires solving an MILP initial relaxation, and the separation problem involves solving the NP-hard maximum s-club problem in the interdicted graph. Second, Constraint (3.3b) will become redundant "easily" if any vertex in S is interdicted. (Contrast this with the clique interdiction counterpart studied by Furini et al. (2019); if S were a clique in G, the constraint would say $\theta \geq |S| - x(S)$ as the clique property is hereditary under vertex deletion.) This behavior can result in weak LP relaxations and it is exacerbated in the presence of numerous "nearly" identical s-clubs, each requiring the addition of a distinct constraint of the form (3.3b) to the initial problem. There is empirical evidence that having a large number of similar s-clubs can be very detrimental for such delayed constraint generation approaches from a computational perspective, especially when the generated constraint is arguably not very strong (Lu et al., 2018; Moradi and Balasundaram, 2018). In the following, we develop techniques that help alleviate the aforementioned concerns by exploiting graph-theoretic properties of s-clubs.

3.4 Exploiting heredity in s-clubs

In this section, we discuss an alternative formulation for the s-club interdiction problem that addresses the issues that arise from using constraints (3.3b) in a delayed constraint generation framework. The formulation is based on the observation that removing vertices of an s-club does not necessarily imply that the remaining vertices do not form an s-club. In other words, the formulation exploits the fact that some s-clubs can be *partially hereditary* in the following sense.

Definition 6. *Given a graph $G = (V, E)$, an s-club S in G, and $H \subseteq S$, we say that S is an H-hereditary s-club if* $\operatorname{diam}(G[S \setminus T]) \leq s$ *for every $T \subseteq H$.*

Observe that every s-club is trivially $\emptyset$-hereditary. Furthermore, an s-club S could be simultaneously H-hereditary and J-hereditary where J and H are incomparable subsets of

S. Therefore, we are only interested in H-hereditary s-clubs of S for which H is maximal with respect to inclusion of vertices from $S \setminus H$. Given an H-hereditary s-club S, we refer to the partition $\{H, S \setminus H\}$ as the H-partition of S and it is said to be non-trivial if $H \neq \emptyset$. Figure 3.2 illustrates this idea using 2-clubs. Pertinently, an s-club S can be S-hereditary (i.e., truly hereditary) if and only if S is a clique.

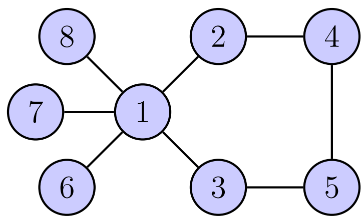

Figure 3.2: The set $\hat{S} = \{1, 2, 3, 4, 5\}$ is a 2-club that admits no non-trivial H-partitions. The 2-club $\tilde{S} = \{1, 2, 3, 6, 7, 8\}$ on the other hand is $\tilde{H}$-hereditary with $\tilde{H} = \{2, 3, 6, 7, 8\}$.

3.5 Alternate formulation using hereditary s-clubs

Given an H-hereditary s-club S, define the following set:

$$\Lambda(S, H) \coloneqq \Big\{(\theta, x) \in \mathbb{R}_+ \times \{0,1\}^{|V|} \mid \theta \geq |S| - x(H) - |S|x(S \setminus H)\Big\}, \tag{3.5}$$

and the following collection of subsets of vertices:

$$\mathcal{C}(S, H) \coloneqq \{S \setminus T \mid T \subseteq H\}. \tag{3.6}$$

In words, $\mathcal{C}(S, H)$ is the collection of all s-clubs generated from S by deleting every possible subset of H and $\mathcal{C}(S, \emptyset) = \{S\}$. The following two lemmas provide the elements that help us to improve Formulation (3.3).

Lemma 1. *Let S be an H-hereditary s-club, and consider an arbitrary point $(\hat{\theta}, \hat{x}) \in \Lambda(S, H)$.*

Then the point $(\hat{\theta}, \hat{x})$ *satisfies the following inequalities:*

$$\theta \geq |U| - |U|x(U) \quad \forall U \in \mathcal{C}(S, H). \tag{3.7}$$

Proof. Consider an arbitrary $U \in \mathcal{C}(S, H)$ and suppose $U = S \setminus T$ for some $T \subseteq H$. Clearly, $S \setminus H \subseteq U$. Suppose $\hat{x}(S \setminus H) \geq 1$. Then, we also have $\hat{x}(U) \geq 1$. By definition of the set $\Lambda(S, H)$ we know that $\hat{\theta} \geq 0$, and hence $(\hat{\theta}, \hat{x})$ satisfies (3.7).

Now suppose $\hat{x}(S \setminus H) = 0$. Then by definition (3.5), the point $(\hat{\theta}, \hat{x})$ satisfies:

$$\hat{\theta} \geq |S| - \hat{x}(H) = |U| + |T| - \hat{x}(H \cap T) - \hat{x}(H \cap U),$$

because U and T partition S which contains H. As $|T| - \hat{x}(H \cap T) \geq 0$, it follows that $(\hat{\theta}, \hat{x})$ satisfies $\hat{\theta} \geq |U| - \hat{x}(H \cap U)$. Again, as $S \setminus H$ and H partition S which contains U, we know that

$$\hat{x}(U) = \hat{x}((S \setminus H) \cap U) + \hat{x}(H \cap U) = \hat{x}(H \cap U),$$

because $\hat{x}(S \setminus H) = 0$. Hence, the point $(\hat{\theta}, \hat{x})$ satisfies $\theta \geq |U| - x(U) \geq |U| - |U|x(U)$ as claimed. □

Based on Lemma 1, when we have two s-clubs U and S such that $U \in \mathcal{C}(S, H)$, we can replace Constraint (3.3b) corresponding to U by the constraint defining the set $\Lambda(S, H)$ in (3.5) without compromising the correctness of Formulation (3.3). Hence, $|\mathcal{C}(S, H)|$ constraints of type (3.3b) can be replaced by a single constraint. For example, the 2-club $\tilde{S} = \{1, 2, 3, 6, 7, 8\}$ in Figure 3.2 is $\tilde{H}$-hereditary for $\tilde{H} = \{2, 3, 6, 7, 8\}$. Hence, we can replace constraints (3.3b) corresponding to all 2-clubs obtained by deleting subsets of $\tilde{H}$ by the single constraint $\theta \geq |\tilde{S}| - x(\tilde{H}) - |\tilde{S}|x(\tilde{S} \setminus \tilde{H})$.

Remark 1. *It is important to contrast the aforementioned discussion against* incorrectly *reformulating* (3.3) *using* $\Lambda(S, H)$*-type constraints only for* s*-clubs that are maximal by* *inclusion. For example, consider the 2-clubs $\hat{S} = \{1, 2, 3, 4, 5\}$ and $\hat{U} = \{2, 4, 5\}$ in Figure 3.2. Although, $\hat{U} \subset \hat{S}$, we know that $\hat{U} \notin \mathcal{C}(\hat{S}, H)$ for any non-empty $H \subseteq \hat{S}$ because $\hat{S}$ does not admit a non-trivial hereditary partition. Therefore, the omission of the constraint $\theta \geq |\hat{U}| - |\hat{U}|x(\hat{U})$ from the formulation would be a mistake because the resulting objective value of the solution defined by $x_v = 1$ for all $v \in V \setminus \hat{U}$ and $x_v = 0$ for all $v \in \hat{U}$ would be zero, rather than the correct objective value of $|\hat{U}|$.*

The notion of H-heredity leads us to consider the following in regards to the strength of the $\Lambda(S, H)$-inequality. If the same s-club S is also J-hereditary, we obtain a different $\Lambda(S, J)$-inequality that is also valid. Is there a particular choice of H that makes the resulting constraint tighter? In this case, maximality of H with respect to the inclusion of vertices from S is the answer. Given an s-club S, we define the set $\mathcal{H}(S)$ as follows:

$$\mathcal{H}(S) := \{H \subseteq S \mid S \text{ is an } H\text{-hereditary } s\text{-club}\}. \tag{3.8}$$

Because $\emptyset \in \mathcal{H}(S)$ for every s-club S in G, by our definition $\mathcal{H}(S)$ is always non-empty.

Lemma 2. *Let S be an s-club such that $H, J \in \mathcal{H}(S)$. If $J \subset H$ then $\Lambda(S, H) \subseteq \Lambda(S, J)$.*

Proof. Let $(\theta, x) \in \Lambda(S, H)$. If $x(S \setminus J) \geq 1$, we have $|S| - x(J) - |S|x(S \setminus J) \leq 0$ and $\theta \geq 0$. Hence, $(\theta, x) \in \Lambda(S, J)$. Now suppose $x(S \setminus J) = 0$. Then, it follows that $x(S \setminus H) = 0$ and $x(H \setminus J) = 0$ as $J \subset H \subseteq S$. Hence, (θ, x) satisfies $\theta \geq |S| - x(H) \geq 0$. Because $x(H \setminus J) = 0$, it also implies that $\theta \geq |S| - x(J) \geq 0$ and $(\theta, x) \in \Lambda(S, J)$, as desired. □

Based on Lemmas 1 and 2, we can replace Constraint (3.3b) for an s-club U with the tighter constraint defining $\Lambda(S, H)$ if $U \in \mathcal{C}(S, H)$, and we only require the constraint for $H \in \mathcal{H}(S)$ that is maximal with respect to inclusion of vertices from S in order to preserve the correctness of the MILP formulation. However, it should be noted that even if the collection $\mathcal{H}(S)$ is limited only to maximal sets, there could be several such maximal elements (see example in Figure 3.3).

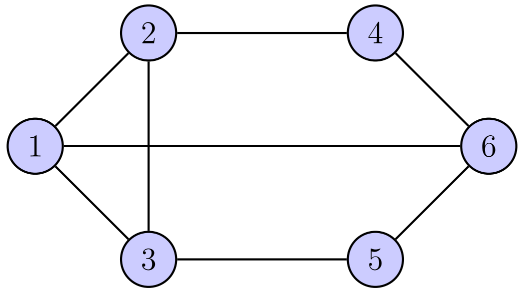

Figure 3.3: $H = \{1\}$ and $J = \{4, 5\}$ are maximal sets in $\mathcal{H}(S) = \{\{1\}, \{4\}, \{5\}, \{4, 5\}\}$ for the 2-club $S = \{1, 2, 3, 4, 5, 6\}$.

Let us define $\mathcal{H}^*(S)$ as the collection of maximal sets in $\mathcal{H}(S)$:

$$\mathcal{H}^*(S) \coloneqq \{H \in \mathcal{H}(S) \mid \text{there is no } J \in \mathcal{H}(S) \text{ such that } H \subset J\}. \tag{3.9}$$

Note that if S does not admit a non-trivial hereditary s-club description (e.g., $\hat{S}$ in Figure 3.2), $\mathcal{H}^*(S) = \mathcal{H}(S) = \{\emptyset\}$. We are now able to state the following result, which is an immediate consequence of the foregoing results and observations.

Lemma 3. *Given an s-club S in $G = (V, E)$, define $\mathcal{U}(S)$ as follows:*

$$\mathcal{U}(S) \coloneqq \bigcup_{H \in \mathcal{H}^*(S)} \mathcal{C}(S, H), \tag{3.10}$$

where $\mathcal{C}(S, H)$ is defined in (3.6)*, and define $\Lambda^*(S)$ as:*

$$\Lambda^*(S) \coloneqq \bigcap_{H \in \mathcal{H}^*(S)} \Lambda(S, H). \tag{3.11}$$

If $(\theta, x) \in \Lambda^*(S)$, then $(\theta, x) \in \Lambda(S, H)$ for all $H \in \mathcal{H}(S)$ and, moreover, (θ, x) satisfies

$$\theta \geq |U| - |U|x(U) \qquad \forall U \in \mathcal{U}(S). \tag{3.12}$$

As a consequence of Lemma 3 we can replace all the constraints in Formulation (3.3) associated with all the s-clubs in $\mathcal{U}(S)$ by $|\mathcal{H}^*(S)|$ stronger constraints to obtain Formulation (3.14) described in Proposition 1 that follows. Depending on the particular s-club, such reduction in the number of constraints can be very significant as illustrated by the following remark.

Remark 2. *A vertex v and its neighbors, i.e., the closed neighborhood $N_G[v]$, is an $N_G(v)$-hereditary s-club for every $s \geq 2$. Every possible subset of $N_G(v)$ is a deletion set T such that $N_G[v] \setminus T$ is an s-club, corresponding to exponentially many constraints in Formulation* (3.3)*. These can all be replaced by a stronger constraint $\theta \geq \deg_G(v)+1-x(N_G(v))-(\deg_G(v)+1)x_v$.*

Proposition 1. *Define $\mathcal{C}^*$, the set of critical s-clubs in the graph $G = (V, E)$, as follows:*

$$\mathcal{C}^* = \{S \in \mathcal{S} \mid \textit{no } s\textit{-club } S' \supset S \textit{ exists such that } S \in \mathcal{U}(S')\}. \tag{3.13}$$

The following is an equivalent reformulation of problem (3.3)*:*

$$z_{s,\alpha} = \min\ \theta + \alpha x(V) \tag{3.14a}$$

$$s.t.\quad \theta \geq |S| - x(H) - |S|x(S \setminus H) \qquad \forall H \in \mathcal{H}^*(S), \forall S \in \mathcal{C}^* \tag{3.14b}$$

$$x \in \{0,1\}^{|V|}, \theta \in \mathbb{R}_+. \tag{3.14c}$$

Proof. We prove that any feasible solution of (3.14) is feasible to (3.3) and vice versa. First, notice that Lemmas 1, 2, and 3 imply that any feasible solution of (3.14) is feasible to (3.3). Now, suppose $(\hat{\theta}, \hat{x})$ is feasible to (3.3), which implies that $\hat{\theta} \geq \bar{\omega}_s(G \setminus T^{\hat{x}}) \geq |S'| - |S'|\hat{x}(S')$ for all $S' \in \mathcal{S}$. Consider $S \in \mathcal{C}^*$ and $H \in \mathcal{H}^*(S)$, chosen arbitrarily, and define $r(S, H, \hat{x}) = |S| - \hat{x}(H) - |S|\hat{x}(S \setminus H)$. Observe that the claim is proven if we can show that $\hat{\theta} \geq r(S, H, \hat{x})$. We consider the following three cases:

(i) $S \subseteq V \setminus T^{\hat{x}}$: No vertex of S is interdicted in this case and hence, $\hat{x}(H) = \hat{x}(S \setminus H) = 0$

and $r(S,H,\hat{x}) = |S|$. Because $S \in \mathcal{S}$, we have that $\hat{\theta} \geq \bar{\omega}_s(G \setminus T^{\hat{x}}) \geq |S|$ and the claim holds.

(ii) $(S \setminus H) \cap T^{\hat{x}} \neq \emptyset$: At least one of the vertices interdicted by $\hat{x}$ belongs to $S \setminus H$. In this case, $\hat{x}(S \setminus H) \geq 1$, which implies that $r(S,H,\hat{x}) \leq 0$, and the claim holds.

(iii) $(S \setminus H) \cap T^{\hat{x}} = \emptyset$ and $H \cap T^{\hat{x}} \neq \emptyset$: Because any vertex in S interdicted by $\hat{x}$ belongs to H, we know that $S \setminus T^{\hat{x}}$ is an s-club in $G \setminus T^{\hat{x}}$, and it follows that $\hat{\theta} \geq \bar{\omega}_2(G \setminus T^{\hat{x}}) \geq |S \setminus T^{\hat{x}}|$. In this case, $r(S,H,\hat{x}) = |S| - |H \cap T^{\hat{x}}| = |S \setminus (H \cap T^{\hat{x}})|$. As $(S \setminus H) \cap T^{\hat{x}} = \emptyset$, we have $S \cap T^{\hat{x}} = H \cap T^{\hat{x}}$ and $S \setminus (H \cap T^{\hat{x}}) = S \setminus T^{\hat{x}}$. Therefore, $r(S,H,\hat{x}) = |S \setminus T^{\hat{x}}|$ and the claim holds.

Thus, we can conclude that any feasible solution of (3.3) is feasible to (3.14). $\square$

Besides having significantly less constraints, Formulation (3.14) does not have redundancies in the sense that all constraints of the form (3.14b) are necessary in the description of the LP relaxation of (3.14); see Proposition 3 in Section 3.5.1.

Two other questions that arise regarding Formulation (3.14) concern the strength of its LP relaxation and whether membership of an s-club in $\mathcal{C}^*$ is easily verifiable. Remark 3 that follows, shows that the LP relaxations of Formulations (3.3) and (3.14) are incomparable. (Hence, both formulations are investigated computationally in Section 4.2.) Proposition 2 that follows provides an alternate characterization of s-clubs in $\mathcal{C}^*$.

Remark 3. *Let P and P' denote the LP relaxations of Formulations* (3.14) *and* (3.3)*, respectively. There are instances where P is not contained in P' and vice versa. In general, for $s \geq 2$, neither LP relaxation contains the other. To see that $P' \nsubseteq P$, consider an s-club $S \in \mathcal{C}^*$ and a non-empty $H \in \mathcal{H}^*(S)$ and construct the point $(\hat{\theta}, \hat{x})$ as follows:*

$$\hat{x}_v = \begin{cases} 1, & \textit{if } v \notin S \\ 0, & \textit{if } v \in S \setminus H \\ 1/2, & \textit{if } v \in H, \end{cases}$$

and $\hat{\theta} = |S| - |H|$. *First we show that* $(\hat{\theta}, \hat{x}) \in P'$. *For any* $U \in \mathcal{S}$, *define* $q(U, x) := |U|(1 - x(U))$, *the right-hand side of Constraint* (3.3b). *If* $U \setminus S$ *is not empty, then* $q(U, \hat{x}) \leq 0$. *On the other hand, if* $U \subseteq S$, *we have* $q(U, \hat{x}) = |U|(1 - |U \cap H|/2)$. *It follows that the maximum value of* $q(U, \hat{x})$ *over* $U \in \mathcal{S}$ *is* $|S| - |H|$, *achieved when* $U = S \setminus H$.

Hence, the point $(\hat{\theta}, \hat{x}) \in P'$. *Furthermore,* $(\hat{\theta}, \hat{x}) \notin P$ *as it violates Constraint* (3.14b) *for the chosen* S *and* H *when* $|H| \geq 1$.

To see that $P \nsubseteq P'$, *we consider a more specific counterexample applicable for any* $s \geq 2$. *Suppose that* $G = (V, E)$ *is a five-vertex star with center 1 and leaves* $\{2,3,4,5\}$. *In this case,* $\mathcal{C}^* = \{V\}$ *with* $\mathcal{H}^*(V) = \{V \setminus \{1\}\}$. *The LP relaxation of Formulation* (3.14) *becomes:*

$$\begin{aligned} &\min\ \theta + \alpha x(V) \\ &\textit{s.t.} \quad \theta \geq 5 - x_2 - x_3 - x_4 - x_5 - 5x_1, \\ &\qquad\quad x \in [0,1]^5, \theta \geq 0. \end{aligned}$$

Consider the point $\bar{\theta} = 13/12$, $\bar{x}_1 = 1/3$, $\bar{x}_2 = 0$, $\bar{x}_3 = \bar{x}_4 = \bar{x}_5 = 1$. *Observe that* $(\bar{\theta}, \bar{x})$ *belongs to* P *but does not belong to* P' *because Formulation* (3.3) *includes the constraint* $\theta \geq 2(1 - x_1 - x_2)$ *corresponding to the s-club* $\{1, 2\}$ *that is violated by* $(\bar{\theta}, \bar{x})$.

Remark 4. *For non-empty* H, *Constraint* (3.14b) *can be tightened using a smaller 'big-M' coefficient as* $\theta \geq |S| - x(H) - (|S| - 1)x(S \setminus H)$ *resulting in a valid formulation with a tighter LP relaxation. However, the conclusion of Remark 3 that the LP relaxations are incomparable continues to hold even using the modified constraint. This can be verified using the same* *counterexamples as in Remark 3. As this modification did not improve the computational performance significantly in our preliminary numerical experiments, we use Constraint* (3.14b) *with the 'big-M' coefficient of* $|S|$ *for simplicity in the subsequent discussions and in our computational studies.*

Another question of interest related to Formulation (3.14) is about the relationship between criticality of an s-club as defined in Proposition 1 and maximality of an s-club (by vertex inclusion). Proposition 2 we establish next shows that maximality is a stricter condition than criticality, that is, every maximal s-club is also a critical s-club although the converse is not true. Consider the example used earlier in Remark 1. The 2-club $\hat{S} = \{1, 2, 3, 4, 5\}$ in Figure 3.2 strictly contains the 2-club $\hat{U} = \{2, 4, 5\}$. The 2-club $\hat{S}$ is both critical and maximal, while $\hat{U}$ is clearly not maximal by inclusion. However, $\hat{U}$ is critical according to the definition in Proposition 1 because $\hat{S}$, which is the *unique* 2-club strictly containing $\hat{U}$, does not admit any non-trivial H-partitions. Indeed, criticality is equivalent to a weaker requirement that we refer to as *one-step maximality* for convenience.

Definition 7. *We say that an s-club S in graph $G = (V, E)$ is one-step maximal if and only if $S \cup \{v\}$ is not an s-club for any $v \in V \setminus S$.*

Observe that if an s-club is maximal then it is also one-step maximal, but the converse is not true. The 2-club $\hat{U}$ is one-step maximal but it is not maximal by inclusion in the conventional sense. It is also easy to see that for cliques and other hereditary properties, one-step maximality is equivalent to inclusionwise maximality.

Proposition 2. *Consider an s-club S in graph $G = (V, E)$. Then, $S \in \mathcal{C}^*$ if and only if S is one-step maximal.*

Proof. Suppose that an s-club U is not critical. Then, there exists another s-club S and a non-empty $H \in \mathcal{H}^*(S)$, such that $U \in \mathcal{C}(S, H)$. In particular, $U = S \setminus T$ for some non-empty $T \subseteq H$ and U is also an s-club because S is an H-hereditary s-club. Now, for a vertex $v \in T$ and consider $U' = U \cup \{v\}$, distinct from U by construction. Note that $U' = (S \setminus T) \cup \{v\} = S \setminus (T \setminus \{v\})$ is also an s-club because $T \setminus \{v\} \subseteq H$ and S is H-hereditary. Then, it follows that U is not one-step maximal.

Conversely, if U is an s-club that is not one-step maximal, then there exists some vertex $v \in V \setminus U$ such that $U \cup \{v\}$ is an s-club. Then, $U \cup \{v\}$ is a $\{v\}$-hereditary s-club. Hence, $U \in \mathcal{C}(U \cup \{v\}, \{v\})$ and is therefore not critical. $\square$

Although deciding if an s-club is maximal by inclusion is coNP-complete (Pajouh and Balasundaram, 2012), Proposition 2 enables us to verify whether a given s-club S is critical in polynomial time. Nonetheless, using Formulation (3.14) directly is not expected to be computationally viable because it requires the enumeration of all s-clubs in $\mathcal{C}^*$ and their maximal hereditary partitions based on $\mathcal{H}^*(\cdot)$. Pertinently, given an s-club S, the complexity of enumerating $\mathcal{H}^*(S)$ or identifying a member in it is also unclear.

However, recall the discussion in Section 3.3 on a delayed constraint generation algorithm. In each iteration i, such a sequential cutting plane method would maintain a collection of s-clubs $\mathcal{S}^i \subset \mathcal{S}$ and for each $S \in \mathcal{S}^i$ it would also maintain collections $\widetilde{\mathcal{H}}(S) \subset \mathcal{H}(S)$. Then, the algorithm solves the following initial relaxation MILP (compare with initial problem (3.4)):

$$z^i_{s,\alpha} = \min_{\substack{x \in \{0,1\}^{|V|} \\ \theta \in \mathbb{R}_+}} \left\{ \theta + \alpha x(V) \;\middle|\; \theta \geq |S| - x(H) - |S| x(S \setminus H) \ \forall S \in \mathcal{S}^i, H \in \widetilde{\mathcal{H}}(S) \right\}. \tag{3.15}$$

Denote the optimal solution found by (θ^i, x^i), we proceed similarly by identifying an s-club S' in the interdicted graph $G \setminus T^{x^i}$ such that $|S'| > \theta^i$, if it exists; otherwise, the solution is feasible and optimal. If found, an important difference is that now, instead of adding the constraint $\theta \geq |S'| - |S'| x(S')$, we will seek to identify a member $H' \in \mathcal{H}^*(S')$ (if that is not possible, find a member $H' \in \mathcal{H}(S')$). Then, we can add the constraint $\theta \geq |S'| - x(H') - |S'| x(S' \setminus H')$, update $\mathcal{S}^{i+1}$ with $\mathcal{S}^i \cup \{S'\}$, update $\widetilde{\mathcal{H}}(S')$ with $\widetilde{\mathcal{H}}(S') \cup \{H'\}$, and then re-solve the initial relaxation. Alternately, we could add a round of constraints by enumerating multiple members of $\mathcal{H}(S')$. Nonetheless, the $\Lambda(S', H)$ inequality is violated by (θ^i, x^i) for every $H \in \mathcal{H}(S')$ as $x^i(S') = 0$; recall that the s-club S' was found in the interdicted graph.

The foregoing discussion highlights the important considerations when separating $\Lambda(S, H)$-inequalities. In particular, how can we detect an $H \in \mathcal{H}^*(S)$? We address this question in Section 3.6. We close this section by discussing polyhedral properties of the LP relaxation and of the convex hull of feasible solutions of Formulation (3.14).

3.5.1 Facial structure of associated polyhedra

Here, we show that the LP relaxation of Formulation (3.14) has no redundant constraints, then we show three types of facets of the convex hull of the formulation based on maximal cliques, critical stars, and critical edge stars of G under an additional assumption of independence among some vertices in the s-club.

First, we discuss the results needed to prove Proposition 3. The LP relaxation P of Formulation (3.14) is full dimensional because $(\theta = |V|, x_v = 1/|V| \colon v \in V) \in \text{interior}(P)$. Consider an $S \in \mathcal{C}^*$ and $H \in \mathcal{H}^*(S)$ that define the face $F(S, H)$ of the polyhedron P given by the corresponding (S, H)-constraint (3.14b), that is,

$$F(S, H) \coloneqq \{(\theta, x) \in P \mid \theta = r(S, H, x)\}\,. \tag{3.16}$$

where we recall that $r(S, H, x) = |S| - x(H) - |S|x(S \setminus H)$.

Lemma 4. *The face $F(S, H)$ in equation* (3.16) *is not contained within any of the following faces of P: $\{(\theta, x) \in P \mid \theta = 0\}$ and $\{(\theta, x) \in P \mid x_v = i\}$ for each $v \in V$ and $i \in \{0, 1\}$.*

Proof. Define $\theta' = |S| - |H|$, $x'_v = 0$ if $v \in S \setminus H$ and $x'_v = 1$ if $v \in (V \setminus S) \cup H$. Note that (θ', x') must belong to P. For any $S' \in \mathcal{C}^*$ and $H' \in \mathcal{H}^*(S')$, if $x'(S' \setminus H') \geq 1$, then

$r(S', H', x') \leq 0 \leq \theta'$. On the other hand if $x'(S' \setminus H') = 0$, then $S' \setminus H' \subseteq S \setminus H$, and $\theta' = |S| - |H| \geq |S'| - |H'| = r(S', H', x')$. As $r(S, H, x') = \theta'$, we know that $(\theta', x') \in F(S, H)$. Moreover, point (θ', x') is not in the $(x_v = 1)$-face of $v \in S \setminus H$ and not in the $(x_v = 0)$-face of $v \in (V \setminus S) \cup H$.

Now consider another point defined as $\tilde{\theta} = |S|$, $\tilde{x}_v = 0$ if $v \in S$ and $\tilde{x}_v = 1$ if $v \in V \setminus S$. Note that $(\tilde{\theta}, \tilde{x}) \in F(S, H)$ based on similar arguments.

The point $(\tilde{\theta}, \tilde{x})$ is not contained in the $(x_v = 1)$-face of $v \in S$, not contained in the $(x_v = 0)$-face of $v \in V \setminus S$, and not contained in the $(\theta = 0)$-face as $|S| \geq 1$. Next we show that $F(S, H)$ can neither belong to the $(x_v = 0)$-face for $v \in S \setminus H$ nor to the $(x_v = 1)$-face for $v \in V \setminus S$ to complete the proof.

For any $U \in \mathcal{C}^*$, $J \in \mathcal{H}^*(U)$, and $x \in [0, 1]^{|V|}$, we know that $r(U, J, \tilde{x}) \leq |S| = \tilde{\theta}$ for any $U \in \mathcal{C}^*$, $J \in \mathcal{H}^*(U)$ as $(\tilde{\theta}, \tilde{x}) \in P$. If in addition $U \neq S$, we claim that $r(U, J, \tilde{x}) \leq |S| - 1$. Indeed, if $U \subset S$ or $U \cap S = \emptyset$ then the claim follows from the definition of $\tilde{x}$. Thus, suppose that $U \cap S \neq \emptyset$ and $U \setminus S \neq \emptyset$.

If $U \cap S \subset S$, i.e., $S \setminus U$ is non-empty, then

$$r(U, J, \tilde{x}) = |U| - |J \setminus S| - |U| \times |(U \setminus J) \setminus S| \leq |U| - |J \setminus S| - |(U \setminus J) \setminus S| = |U \cap S| \leq |S| - 1.$$

Now suppose $S \subset U$. We also know that $U \setminus J \neq \emptyset$, as otherwise U is a clique that contains S, which contradicts $S \in \mathcal{C}^*$. If in addition, $(U \setminus J) \cap (V \setminus S) = \emptyset$, it follows that $U \setminus J \subseteq S$. Consider the following relationships: $U \setminus J \subseteq S \subset U$, which implies that $U \setminus S \subseteq J$. Hence, S can be obtained from U by deleting $U \setminus S \in \mathcal{H}(U)$, a contradiction to $S \in \mathcal{C}^*$. Therefore, if $S \subset U$ it must be the case that $(U \setminus J) \cap (V \setminus S) \neq \emptyset$. If so, we obtain $r(U, J, \tilde{x}) \leq 0 \leq |S| - 1$, as desired. So the claim holds.

We are now ready to demonstrate a point $(\hat{\theta}, \hat{x}) \in F(S, H)$ that is not contained in the $(x_v = 0)$-face for an arbitrarily chosen $v \in S \setminus H$. Define $\hat{x}_u = \tilde{x}_u \ \forall u \neq v$ and with $\hat{x}_v = 1/|S|$;

let $\hat{\theta} = |S| - 1$. Then, $r(S, H, \hat{x}) = |S| - 1$ and therefore $(\hat{\theta}, \hat{x})$ satisfies the equality constraint in $F(S, H)$. On the other hand, as $\hat{x} > \tilde{x}$ we obtain $r(U, J, \hat{x}) \leq r(U, J, \tilde{x})$ for any $U \in \mathcal{C}^*$ and $J \in \mathcal{H}^*(U)$. Therefore, as $r(U, J, \tilde{x}) \leq |S| - 1 = \hat{\theta}$, we conclude that $\hat{\theta} \geq r(U, J, \hat{x})$ for any $U \in \mathcal{C}^*$ and $J \in \mathcal{H}^*(U)$. In other words, $(\hat{\theta}, \hat{x})$ belongs to $F(S, H)$ but it does not belong into the face of P induced by $x_v = 0$.

Now we demonstrate a point $(\bar{\theta}, \bar{x}) \in F(S, H)$ that is not contained in the $(x_v = 1)$-face for an arbitrarily chosen $v \in V \backslash S$. Consider the same $\tilde{x}$ as in the previous case and define $\bar{x}_u = \tilde{x}_u$ for each $u \neq v$ and set $\bar{x}_v = 1 - \epsilon$, where the positive constant $\epsilon < 1/|V|$, and let $\bar{\theta} = \tilde{\theta} = |S|$. Observe that $r(S, H, \bar{x}) = r(S, H, \tilde{x})$ and therefore $(\bar{\theta}, \bar{x})$ satisfies the constraint defining $F(S, H)$ at equality. Similarly, $r(U, J, \bar{x}) = r(U, J, \tilde{x})$ for any s-club U that does not contain vertex v. Hence, if $v \notin U$, $(\bar{\theta}, \bar{x})$ satisfies the corresponding constraint $\theta \geq r(U, J, x)$. If $v \in J$, then $r(U, J, \bar{x}) = r(U, J, \tilde{x}) + \epsilon \leq |S| - 1 + 1/|V| < |S| = \bar{\theta}$, thus $(\bar{\theta}, \bar{x})$ satisfies the constraint $\theta \geq r(U, J, x)$. Finally, if $v \in U \setminus J$, then $r(U, J, \bar{x}) = r(U, J, \tilde{x}) + \epsilon|U| \leq |S| - 1 + 1 = |S| = \bar{\theta}$. Again $(\bar{\theta}, \bar{x})$ satisfies the constraint $\theta \geq r(U, J, x)$ if $v \in U \setminus J$. Hence, $(\bar{\theta}, \bar{x}) \in F(S, H)$, but it does not belong to the face induced by $x_v = 1$. Hence, $F(S, H)$ is not contained within any of the trivial faces of P. □

Proposition 3. *Every Constraint* (3.14b) *induces a facet of the LP relaxation polyhedron of* (3.14).

Proof. Consider $\hat{S} \in \mathcal{C}^*$ and $\hat{H} \in \mathcal{H}^*(\hat{S})$ also chosen arbitrarily such that $(S, H) \neq (\hat{S}, \hat{H})$. We claim that there exists a point $(\tilde{\theta}, \tilde{x}) \in F(S, H)$ such that $(\tilde{\theta}, \tilde{x}) \notin F(\hat{S}, \hat{H})$; this assertion in conjunction with Lemma 4 would yield the desired result. This is because, if $F(S, H) \setminus F(\hat{S}, \hat{H}) \neq \emptyset$, we know that the face $F(S, H)$ cannot be completely contained in the face $F(\hat{S}, \hat{H})$. Since the latter is arbitrary, it shows that no other inequality (3.14b) induces a face of P that contains $F(S, H)$. Therefore, $F(S, H)$ must be maximal.

First, we assume that $\hat{S} = S$. It then follows that $\hat{H} \neq H$, which in turn implies that

$S \setminus H \neq \emptyset$; recall that if $S = H$, then $S = \hat{S}$ must be a clique, in which case $\mathcal{H}^*(\hat{S}) = \{\hat{S}\}$ as it only contains maximal members. Consider the point constructed as follows: $\tilde{\theta} = |S| - |H|$, $\tilde{x}_v = 0$ if $v \in S \setminus H$ and $\tilde{x}_v = 1$ if $v \in (V \setminus S) \cup H$. Note that $(\tilde{\theta}, \tilde{x}) \in F(S, H)$ as the defining inequality is active at $(\tilde{\theta}, \tilde{x})$ and the point belongs to P (easy to verify). Now, because $H, \hat{H} \in \mathcal{H}^*(S)$, then H is not contained in $\hat{H}$ and vice versa. This observation implies that $S \setminus \hat{H}$ is not contained in $S \setminus H$, consequently $\tilde{x}(S \setminus \hat{H}) \geq 1$. Therefore, $|\hat{S}| - \tilde{x}(\hat{H}) - |\hat{S}|\tilde{x}(\hat{S} \setminus \hat{H}) \leq 0$ and $\tilde{\theta} > 0$, which implies that $(\tilde{\theta}, \tilde{x}) \notin F(\hat{S}, \hat{H})$.

Now we assume that $S \neq \hat{S}$ and consider the following point: $\tilde{\theta} = |S|$, $\tilde{x}_v = 0$ if $v \in S$ and $\tilde{x}_v = 1$ if $v \in V \setminus S$. Note that $(\tilde{\theta}, \tilde{x}) \in F(S, H)$. Suppose that $\hat{S} \setminus \hat{H}$ is not contained in S. Then, $\tilde{x}(\hat{S} \setminus \hat{H}) \geq 1$ and therefore $|\hat{S}| - \tilde{x}(\hat{H}) - \tilde{x}(\hat{S} \setminus \hat{H}) \leq 0$, which implies that $(\tilde{\theta}, \tilde{x}) \notin F(\hat{S}, \hat{H})$ as $\tilde{\theta} = |S| \geq 1$. Next consider the case where $\hat{S} \setminus \hat{H} \subseteq S$. Because $S, \hat{S} \in \mathcal{C}^*$, by the definition of $\mathcal{C}^*$ we know that $\hat{S} \setminus \hat{H} \neq S$; hence, the containment must be strict. Now, partition $\hat{H}$ as $\hat{H} = \hat{H}_1 \cup \hat{H}_2$, where $\hat{H}_1 = \hat{H} \setminus S$ and $\hat{H}_2 = \hat{H} \cap S$. From the fact that $\hat{S} \setminus \hat{H} \subset S$, it follows that the right-hand side of the constraint inducing face $F(\hat{S}, \hat{H})$ evaluated at $(\tilde{\theta}, \tilde{x})$ becomes:

$$|\hat{S}| - \tilde{x}(\hat{H}) - \tilde{x}(\hat{S} \setminus \hat{H}) = |\hat{S}| - \tilde{x}(\hat{H}) = |\hat{S}| - |\hat{H}_1|.$$

Now, we claim that $|\hat{S}| - |\hat{H}_1| < |S|$. Suppose, for the sake of contradiction that this is not the case, i.e, $|\hat{S}| - |\hat{H}_1| = |S|$. Then, as $\hat{S} \setminus \hat{H}_1 \subseteq S$ this would imply that $\hat{S} \setminus \hat{H}_1 = S$. However, this would contradict the definition of $\mathcal{C}^*$ as $S \in \mathcal{C}^*$. Therefore, $|\hat{S}| - |\hat{H}_1| < |S| = \tilde{\theta}$ implying that $(\tilde{\theta}, \tilde{x}) \notin F(\hat{S}, \hat{H})$, which completes the proof. □

The result in Proposition 3 indicates the importance of critical s-clubs in $\mathcal{C}^*$ (and the maximal members in $\mathcal{H}^*(S)$ for every critical s-club S) in formulating this problem. It further emphasizes the fact that no constraint of type (3.14b) is dominated by another of this type in the associated LP relaxation. This result also motivates the facets of the *convex hull* of feasible solutions to Formulation (3.14) we derive based on specially structured s-clubs. These results are presented next.

Let $\mathcal{P}$ denote the convex hull of the set of feasible solutions of Formulation (3.14). As it is to be expected, constraints (3.14b) do not yield facets of $\mathcal{P}$ in general because of the 'big-M' type constant $|S|$ in the constraint. To identify facets of $\mathcal{P}$, we begin with an inequality that is known to induce a facet of the clique interdiction counterpart. Furini et al. (2019) formulate the clique interdiction problem (with an interdiction budget instead of a penalty) using the following constraints:

$$\theta \geq |K| - x(K) \quad \forall K \in \mathcal{K}, \tag{3.17}$$

where $\mathcal{K}$ is the collection of all cliques in G. Because the clique property is hereditary, there is no need for a 'big-M' coefficient in Constraint (3.17) to make the constraint redundant if a vertex in K is interdicted. Furini et al. (2019) further show that inequality (3.17) induces a facet of the convex hull of feasible solutions to their budget-constrained clique interdiction problem under suitable conditions, one of which is the maximality of clique K.

Cliques are s-clubs for every $s \geq 2$ and remain so if some vertices are interdicted. So the inequality (3.17) is valid for the s-club interdiction problem as well, and it is reasonable to ask if they induce facets when the clique K satisfies some additional requirement (like maximality). Next, we provide a result that generalizes these facets to s-clubs, for any $s \geq 2$.

For a given subset of vertices $Q \subseteq V$, let $\mathcal{P}^Q$ denote the face of the convex hull $\mathcal{P}$ in which the vertices of Q are not interdicted, that is, $\mathcal{P}^Q = \mathcal{P} \cap \{(\theta, x) \mid x_v = 0 \ \forall v \in Q\}$ (see Lemmas 5 and 6 regarding the "zero facets" of $\mathcal{P}$). One can consider $\mathcal{P}^Q$ as the convex hull of interest at a node of a branch-and-cut tree where the variables corresponding to Q have been fixed to zero. However, we are more interested in the case where $Q = S \setminus H$ for some H-hereditary s-club S when certain facets of $\mathcal{P}^Q$ can be readily derived, as shown in

Theorem 1.

Lemma 5. *Consider a non-empty graph $G = (V, E)$ and positive integer s. The convex hull of feasible solutions to Formulation* (3.14)*, denoted by $\mathcal{P}$, is full dimensional.*

Proof. Let e_v denote the $|V|$-dimensional unit vector with v-th component at one. It is easy to verify that the following $|V| + 2$ points in $\mathcal{P}$, $(\theta = |V|, x = e_v) \in \mathcal{P}$ for each $v \in V$, $(\theta = |V|, x = \mathbf{0})$, and $(\theta = 0, x = \mathbf{1})$, are affinely independent. □

Lemma 6. *Consider a non-empty graph $G = (V, E)$ and positive integer s. The valid inequality $x_v \geq 0$ induces a facet of $\mathcal{P}$ for each $v \in V$.*

Proof. Let the "zero face" corresponding to vertex v be denoted as $F_v \coloneqq \{(\theta, x) \in \mathcal{P} \mid x_v = 0\}$. As $\dim(\mathcal{P}) = |V|+1$, we demonstrate the same number of affinely independent points contained in F_v to establish this claim. The following can be easily verified as affinely independent points: $(\theta = |V|, x = e_u) \in \mathcal{P}$ for each $u \in V \setminus \{v\}$, $(\theta = |V|, x = \mathbf{0})$, and $(\theta = 1, x = \mathbf{1} - e_v)$. □

Theorem 1. *Let $S \in \mathcal{C}^*$ be an H-hereditary s-club. Then the following inequality is valid for $\mathcal{P}^{S\setminus H}$ and induces a facet of $\mathcal{P}^{S\setminus H}$ for any positive integer s:*

$$\theta \geq |S| - x(H). \tag{3.18}$$

Proof. Consider an s-club $S \in \mathcal{C}^*$ and an $H \in \mathcal{H}(S)$. Note that $S \setminus H$ may be empty if $S = H$ is a clique. We know from Lemma 6 that $\mathcal{P}^{S\setminus H} \coloneqq \mathcal{P} \cap \{(\theta, x) \mid x_v = 0 \; \forall v \in S \setminus H\}$ is a face of $\mathcal{P}$, and hence,

$$\dim(\mathcal{P}^{S\setminus H}) = \dim(\mathcal{P}) - |S \setminus H|.$$

The inequality $\theta \geq |S| - x(H)$ is valid for $\mathcal{P}^{S\setminus H}$ because $x_v = 0$ for every $v \in S \setminus H$ and S is H-hereditary. The following collection of $\dim(\mathcal{P}^{S\setminus H})$ points can be easily verified to be contained in the face,

$$F_Q := \{(\theta, x) \in \mathcal{P}^{S\setminus H} \mid \theta = |S| - x(H)\}.$$

1. Construct the first $|H|$ points $(\hat{\theta}, \hat{x})$ for every vertex $u \in H$ where $\hat{\theta} = |S| - 1$ and $\hat{x}$ is defined as
$$\hat{x}_v = \begin{cases} 1, & \text{if } v = u, \\ 1, & \text{if } v \in V \setminus S, \\ 0, & \text{if } v \in S \setminus \{u\}. \end{cases}$$

2. Construct the next $|V \setminus S|$ points $(\hat{\theta}, \hat{x})$ for every vertex $u \in V \setminus S$ where $\hat{\theta} = |S|$ and $\hat{x}$ is defined as
$$\hat{x}_v = \begin{cases} 0, & \text{if } v \in S \cup \{u\} \\ 1, & \text{if } v \in V \setminus (S \cup \{u\}). \end{cases}$$
Note that because S is a critical s-club, $S \cup \{u\}$ cannot be an s-club based on Proposition 2.

3. Finally consider the point, $(\hat{\theta}, \hat{x})$ where $\hat{\theta} = |S|$ and $\hat{x}$ defined as
$$\hat{x}_v = \begin{cases} 0, & \text{if } v \in S \\ 1, & \text{if } v \in V \setminus S. \end{cases}$$

The foregoing $\dim(\mathcal{P}^{S\setminus H}) = |V|+1-|S|+|H|$ points can be verified to be affinely independent, establishing our claim. □

Although H is not required to be a maximal member of $\mathcal{H}(S)$ for Theorem 1 to hold, it is relevant in the following sense. Such an inequality is valid (without lifting the variables in $S \setminus H$) only locally in the nodes of a branch-and-cut tree where the corresponding variables have been fixed to zero. It could therefore be argued that larger $H \in \mathcal{H}(S)$ will make this inequality usable higher up in the branch-and-cut tree where it could be even more effective.

This observation leads us to consider the special case $S = H$, where (3.18) is valid for $\mathcal{P}$ and induces a facet if $S \in \mathcal{C}^*$. Recall from the discussions following Definition 6 that S is S-hereditary only if it is a clique, in which case inequality (3.18) is precisely inequality (3.17) for clique S. If $s = 1$ and we consider clique interdiction, this inequality induces a facet of $\mathcal{P}$ if the clique $S \in \mathcal{C}^*$. We also know from Proposition 2 that the 1-club (clique) $S \in \mathcal{C}^*$ if and only if it is one-step maximal. As clique is a hereditary property, this is equivalently saying that the clique S must be inclusionwise maximal for inequality (3.18) to induce a facet of $\mathcal{P}$. Therefore, the special case of Theorem 1 with $H = S$ and $s = 1$ extends the result of Furini et al. (2019) to our setting with interdiction penalty.

Now consider the same special case $S = H$ but with $s \geq 2$. For a clique S to be a critical s-club, i.e., a one-step maximal s-club, no vertex in $V \setminus S$ can be adjacent to a vertex in S; otherwise, such a vertex along with vertices in S forms an s-club for any $s \geq 2$. Thus, we can conclude that if S is clique that induces a maximal connected component of the graph G, inequality (3.18) induces a facet of $\mathcal{P}$ for any $s \geq 2$. We can now see Theorem 1 as a generalization of the result of Furini et al. (2019) to s-club interdiction under interdiction penalty *for any* $s \geq 2$. It should be noted, however, that the criticality requirement on the clique is a very restrictive condition when $s \geq 2$, as the clique must form a connected component by itself. It turns out, as the following theorem established by a direct proof shows, that it is sufficient for the clique S to be maximal with respect to the clique property (and not necessarily critical with respect to the s-club property) for inequality (3.17) to induce a facet of $\mathcal{P}$ for any $s \geq 2$.

Theorem 2. *Given a graph $G = (V, E)$, a positive integer s, and an inclusionwise maximal* *clique S in G, the following inequality is valid for $\mathcal{P}$ and induces a facet of $\mathcal{P}$:*

$$\theta \geq |S| - x(S). \tag{3.19}$$

Proof. Validity of inequality (3.19) is easy to see as the clique S is hereditary under vertex deletion and it is an s-club for every $s \geq 2$. We prove that the face F' of $\mathcal{P}$ induced by inequality (3.19) is maximal. That is,

$$F' := \{(\theta, x) \in \mathcal{P} \mid \theta + x(S) = |S|\}.$$

Consider an arbitrary proper face of $\mathcal{P}$ given by:

$$F := \{(\theta, x) \in \mathcal{P} \mid a_0\theta + \sum_{i \in V} a_i x_i = b\},$$

which we assume contains F' in order to arrive at a contradiction.

Consider the following point: $\theta = |S|; x_u = 1\ \forall u \notin S; x_u = 0\ \forall u \in S$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$|S|a_0 + \sum_{i \notin S} a_i = b. \tag{3.20}$$

Now consider the following point for some $\ell \in S$: $\theta = |S| - 1; x_u = 1\ \forall u \in (V \setminus S) \cup \{\ell\}; x_u = 0\ \forall u \in S \setminus \{\ell\}$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(|S| - 1)a_0 + \sum_{i \notin S} a_i + a_\ell = b. \tag{3.21}$$

From equations (3.20) and (3.21), we can conclude that $a_\ell = a_0\ \forall \ell \in S$ and rewrite F as follows:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0x(S) + \sum_{i \notin S} a_i x_i = b\}. \tag{3.22}$$

Finally, consider the following point for an arbitrary vertex $\ell \notin S$: $\theta = |S \setminus N_G(\ell)|$; $x_u = 1\ \forall u \in N_G(\ell) \cup [V \setminus (S \cup \{\ell\})]$; $x_u = 0\ \forall u \in \{\ell\} \cup S \setminus N_G(\ell)$. Because S is a *maximal* clique, vertex ℓ cannot be adjacent to every vertex in S. Hence, we know that $S \setminus N_G(\ell)$ is a *non-empty* clique. We also know that $\{\ell\} \cup S \setminus N_G(\ell)$ is not an s-club as vertex ℓ is isolated in the graph interdicted according to x. As $\ell \notin S$, we know that $x(S) = |S \cap N_G(\ell)|$, and hence $\theta + x(S) = |S|$, implying that $(\theta, x) \in F'$. Now, using Equation (3.22) we can obtain the following equation as $F' \subseteq F$:

$$a_0|S \setminus N_G(\ell)| + a_0|S \cap N_G(\ell)| + \sum_{i \notin S \cup \{\ell\}} a_i = b. \tag{3.23}$$

From equations (3.20) and (3.23), we can conclude that $a_\ell = 0$ for each $\ell \notin S$ and $b = |S|a_0$. We now know that F has the following form:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0x(S) = |S|a_0\}.$$

As F is a proper face, we know that $a_0 \neq 0$ and we can conclude that $F' = F$ is a maximal proper face. $\square$

Because enumerating maximal cliques is not computationally desirable given that there could be exponentially many in a graph (Moon and Moser, 1965), we do not explicitly make use of this facet in our computational studies. However, the next two results—based on specially structured s-clubs—are interesting to us from a computational perspective.

Theorem 3. *Given a graph $G = (V, E)$ and an integer $s \geq 2$, suppose that for some vertex $v \in V$ the closed neighborhood of v forms a critical s-club. That is, $N_G[v] \in \mathcal{C}^*$, a critical star* *centered at v. If $N_G(v)$ is an independent set, the following inequality is valid and induces a facet of $\mathcal{P}$:*

$$\theta \geq \deg_G(v) + 1 - x(N_G(v)) - \deg_G(v)x_v. \tag{3.24}$$

This inequality can be viewed as a strengthening of the coefficient of x_v in Constraint (3.14b) with $S = N_G[v]$ and $H = N_G(v)$.

Proof. Validity of inequality (3.24) follows from the observation that for any $x \in \{0,1\}^{|V|}$ and $\theta \in \mathbb{R}$, we know that $(\theta, x) \in \mathcal{P}$ if and only if $\theta \geq \bar{\omega}_s(G \setminus T^x)$. We know that if $x_v = 0$, $\bar{\omega}_s(G \setminus T^x) \geq \deg_G(v) + 1 - x(N_G(v))$ and the inequality is valid. If $x_v = 1$ and $N_G(v)$ is an independent set, the inequality imposes that $\theta \geq 1 - x(N_G(v))$, which is valid for all $x \in \mathcal{P}$.

As before, we show that inequality (3.24) induces a maximal face of $\mathcal{P}$. We define,

$$F' := \{(\theta, x) \in \mathcal{P} \mid \theta + x(N_G(v)) + d_v x_v = d_v + 1\},$$

where $d_v \equiv \deg_G(v)$. Consider an arbitrary proper face of $\mathcal{P}$ given by:

$$F := \{(\theta, x) \in \mathcal{P} \mid a_0\theta + \sum_{i \in V} a_i x_i = b\},$$

which we assume contains F'.

Consider the following point: $\theta = d_v + 1; x_u = 1\ \forall u \notin N_G[v]; x_u = 0\ \forall u \in N_G[v]$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(d_v + 1)a_0 + \sum_{i \notin N_G[v]} a_i = b. \tag{3.25}$$

Now consider the following point for some $\ell \in N_G(v)$: $\theta = d_v; x_u = 1\ \forall u \notin N_G[v]; x_\ell =$

$1; x_u = 0\ \forall u \in N_G[v] \setminus \{\ell\}$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$d_v a_0 + \sum_{i \notin N_G[v]} a_i + a_\ell = b. \tag{3.26}$$

From equations (3.25) and (3.26), we can conclude that $a_\ell = a_0\ \forall \ell \in N_G(v)$ and rewrite F as follows:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0 x(N_G(v)) + a_v x_v + \sum_{i \notin N_G[v]} a_i x_i = b\}.$$

Next we consider an arbitrary $\ell \notin N_G[v]$. As $N_G[v] \in \mathcal{C}^*$ is a one-step maximal (critical) s-club, we know that $N_G[v] \cup \{\ell\}$ cannot be an s-club; otherwise, we will contradict the definition of $\mathcal{C}^*$ (see Proposition 2). Hence, we consider the following point: $\theta = d_v + 1; x_u = 0\ \forall u \in N_G[v] \cup \{\ell\}; x_u = 1\ \forall u \notin N_G[v] \cup \{\ell\}$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(d_v + 1)a_0 + \sum_{i \notin N_G[v] \cup \{\ell\}} a_i = b. \tag{3.27}$$

Now from equations (3.25) and (3.27), we can conclude that $a_\ell = 0$ for each $\ell \notin N_G[v]$ and $b = (d_v + 1)a_0$. We now know that F has the following form:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0 x(N_G(v)) + a_v x_v = (d_v + 1)a_0\}.$$

Finally to identify the coefficient a_v, we consider the following point: $\theta = 1; x_u = 0\ \forall u \in N_G(v); x_u = 1\ \forall u \notin N_G(v)$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$a_0 + a_v = (d_v + 1)a_0. \tag{3.28}$$

Hence, $a_v = d_v a_0$. Because F is a proper face, we know that $a_0 \neq 0$ and thus we conclude that $F' = F$ is a maximal proper face. $\square$

Theorem 4 that follows is similar to Theorem 3, and is based instead on critical *edge stars*, i.e., sets of the form $N_G(u) \cup N_G(v)$ where $\{u, v\} \in E$. Due to the asymmetry in the coefficients of x_u and x_v, in general, Theorem 4 corresponds to two facet-inducing inequalities obtained by interchanging vertices u and v.

Theorem 4. *Given a graph $G = (V, E)$ and an integer $s \geq 3$, consider adjacent vertices u and v such that $N_G(u) \cup N_G(v) \setminus \{u, v\}$ is a non-empty independent set. If $N_G(u) \cup N_G(v) \in \mathcal{C}^*$, then*

$$\theta \geq \deg_G(u) + \deg_G(v) - c_{uv} - x(L_{uv}) - [\deg_G(u) - c_{uv}]x_u - [\deg_G(v) - \min(c_{uv}, 1)]x_v \tag{3.29}$$

is valid and induces a facet of $\mathcal{P}$, where $L_{uv} := N_G(u) \cup N_G(v) \setminus \{u, v\}$ and $c_{uv} := |N_G(u) \cap N_G(v)|$.

Proof. Validity of inequality (3.29) follows from the observation that for any feasible solution $(\theta, x) \in \mathcal{P}$ of Formulation (3.14), $\theta \geq \bar{\omega}_s(G \setminus T^x)$. If $x_u = x_v = 0$, we know that $\bar{\omega}_s(G \setminus T^x) \geq \deg_G(u) + \deg_G(v) - c_{uv} - x(L_{uv})$ and the inequality is satisfied. If $x_u = x_v = 1$, we require $\theta \geq \min\{1, c_{uv}\} - x(L_{uv})$ which holds.

If $x_u = 1$ and $x_v = 0$, then $\theta \geq \deg_G(v) - x(L_{uv})$ which is valid. Finally, if $x_u = 0$ and $x_v = 1$, we require $\theta \geq \deg_G(u) - c_{uv} - x(L_{uv}) + \min\{1, c_{uv}\}$. Here, we consider two cases. If $c_{uv} = 0$, the inequality becomes $\theta \geq \deg_G(u) - x(L_{uv})$, and if $c_{uv} \geq 1$, the inequality becomes $\theta \geq \deg_G(u) - c_{uv} - x(L_{uv}) + 1$ and the inequality is valid in both cases.

Next, we show that the face of $\mathcal{P}$ induced by inequality (3.29) is maximal.

Let F' denote the face of $\mathcal{P}$ induced by inequality (3.29). That is,

$$F' := \{(\theta, x) \in \mathcal{P} \mid \theta + x(L_{uv}) + (d_u - c_{uv})x_u + (d_v - \min\{1, c_{uv}\})x_v = d_u + d_v - c_{uv}\},$$

where $d_u \equiv \deg_G(u)$. Consider an arbitrary proper face of $\mathcal{P}$ given by:

$$F \coloneqq \{(\theta, x) \in \mathcal{P} \mid a_0\theta + \sum_{i \in V} a_i x_i = b\},$$

which we assume contains F'.

Consider the following point: $\theta = d_u + d_v - c_{uv}; x_i = 1\ \forall i \notin N(u) \cup N(v); x_i = 0\ \forall i \in N(u) \cup N(v)$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(d_u + d_v - c_{uv})a_0 + \sum_{i \notin N(u) \cup N(v)} a_i = b. \tag{3.30}$$

Now for some $\ell \in L_{uv}$, consider the following point: $\theta = d_u + d_v - c_{uv} - 1; x_i = 1\ \forall i \notin N(u) \cup N(v); x_\ell = 1; x_i = 0\ \forall i \in N(u) \cup N(v) \setminus \{l\}$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(d_u + d_v - c_{uv} - 1)a_0 + \sum_{i \notin N(u) \cup N(v)} a_i + a_\ell = b. \tag{3.31}$$

From equations (3.30) and (3.31), we can conclude that $a_\ell = a_0\ \forall \ell \in L_{uv}$ and rewrite F as follows:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0 x(L_{uv}) + a_u x_u + a_v x_v + \sum_{i \notin N(u) \cup N(v)} a_i x_i = b\}.$$

Next we consider an arbitrary $\ell \notin N(u) \cup N(v)$. As $N(u) \cup N(v) \in \mathcal{C}^*$, we know that $N(u) \cup N(v) \cup \{\ell\}$ cannot be an s-club; otherwise, we will contradict the definition of $\mathcal{C}^*$ (see Proposition 2). Hence, we consider the following point: $\theta = d_u + d_v - c_{uv}; x_i = 0\ \forall i \in N(u) \cup N(v) \cup \{\ell\}; x_i = 1\ \forall i \notin N(u) \cup N(v) \cup \{\ell\}$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$(d_u + d_v - c_{uv})a_0 + \sum_{i \notin \{N(u) \cup N(v)\} \cup \{\ell\}} a_i = b. \tag{3.32}$$

Now from equations (3.30) and (3.32), we can conclude that $a_\ell = 0$ for each $\ell \notin N(u) \cup N(v)$ and $b = (d_u + d_v - c_{uv})a_0$. We now know that F has the following form:

$$F = \{(\theta, x) \in \mathcal{P} \mid a_0\theta + a_0 x(L_{uv}) + a_u x_u + a_v x_v = (d_u + d_v - c_{uv})a_0\}.$$

To identify the coefficients a_u and a_v, we consider the following two cases.

(i) Suppose $c_{uv} = 0$. Consider the point $\theta = d_v; x_i = 0\ \forall i \in N(u) \cup N(v) \setminus \{u\}; x_i = 1\ \forall i \notin L_{uv}; x_u = 1$ As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$d_v a_0 + a_u = (d_u + d_v)a_0. \tag{3.33}$$

Hence, $a_u = d_u a_0$.

Now, consider the following point to determine coefficient a_v: $\theta = d_u; x_i = 0\ \forall i \in N(u) \cup N(v) \setminus \{v\}; x_i = 1\ \forall i \notin L_{uv}; x_v = 1$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$d_u a_0 + a_v = (d_u + d_v)a_0. \tag{3.34}$$

Hence, $a_v = d_v a_0$.

(ii) Suppose $c_{uv} \geq 1$. Consider the point: $\theta = d_v; x_i = 0\ \forall i \in N(u) \cup N(v) \setminus \{u\}; x_i = 1\ \forall i \notin L_{uv}; x_u = 1$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$d_v a_0 + a_u = (d_u + d_v - c_{uv})a_0. \tag{3.35}$$

Hence, $a_u = (d_u - c_{uv})a_0$.

Finally, consider the following point to determine coefficient a_v: $\theta = 1; x_i = 0\ \forall i \in L_{uv}; x_i = 1\ \forall i \notin L_{uv}; x_u = x_v = 1$. As $(\theta, x) \in F' \subseteq F$, we obtain the following equation:

$$a_0 + (d_u - c_{uv})a_0 + a_v = (d_u + d_v - c_{uv})a_0. \tag{3.36}$$

Hence, $a_v = (d_v - 1)a_0$.

Combining the two cases together we obtain $a_u = (d_u - c_{uv})a_0$ and $a_v = (d_v - \min\{1, c_{uv}\})a_0$. Because F is a proper face, we know that $a_0 \neq 0$ and thus we conclude that $F' = F$ is a maximal proper face. □

Many real-life social and biological networks demonstrate a power law degree distribution and are also extremely sparse in terms of edge density (Chung and Lu, 2006; Newman, 2003; Barabási and Albert, 1999). So it is not uncommon in practice to find vertices and edges with a large number of independent neighbors in sparse real-life graphs, such as those used in our computational study. Nonetheless, we also do not recommend strictly testing the satisfaction of the sufficient conditions in order to add the critical vertex and edge star facets during computations. These two results essentially serve to motivate our emphasis on vertex and edge stars in building the initial relaxation of Formulation (3.14) used in our delayed constraint generation algorithm discussed in Section 4.1.

3.6 Hereditary s-clubs and latency-s connected dominating sets

Given a graph $G = (V, E)$, we say that $D \subseteq V$ is a *dominating set* if every vertex outside D has a neighbor in D. We say that D is a *connected* dominating set if in addition, $G[D]$ is connected. In essence, a connected dominating set ensures that every pair of distinct vertices outside the dominating set have a path connecting them (whose internal vertices are contained in the dominating set). The connection to hereditary s-clubs, while not obvious, arises when we require distance bounds in addition to connected domination. Definition 8 below is adapted from its counterpart for directed graphs introduced by Validi and Buchanan (2020).

Definition 8 (Validi and Buchanan (2020))**.** *Given a graph $G = (V, E)$, a subset of vertices D is called a latency-s connected dominating set (latency-s CDS) if it is a dominating set in G and for every pair of distinct vertices in V there exists a path of length at most s whose internal vertices belong to D.*

If D is a latency-s CDS, then it is a dominating set that is also an s-club. Note that the length-bounded path requirement applies to vertex-pairs inside D as well. Clearly, a dominating s-club is not necessarily a latency-s CDS (see Figure 3.4a). It is also easy to see that a dominating $(s-2)$-club is a latency-s CDS. However, a latency-s CDS is not necessarily a dominating $(s-2)$-club (see Figure 3.4b).

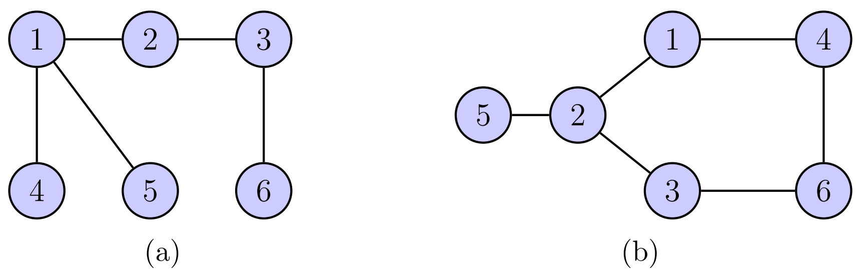

Figure 3.4: (a) Set $\{1, 2, 3\}$ forms a dominating 2-club, but it is not a latency-2 CDS since the length of the path between vertices 5 and 6 is 4. (b) Set $\{1, 2, 3\}$ forms a latency-3 CDS. (Note that vertices 4 and 6 are adjacent and vacuously satisfy the requirement.) Clearly, it is not a 1-club (clique).

Given an s-club S in graph $G = (V, E)$, we say that D is a "latency-s CDS over S" if and only if D is a latency-s CDS in the induced subgraph $G[S]$. In general, a graph G has a latency-s CDS if and only if $\text{diam}(G) \leq s$. The necessity can be deduced from the fact that every pair of vertices must be connected by a path of length at most s, in order for a latency-s CDS to exist. Conversely, if $\text{diam}(G) \leq s$ we know that V is a latency-s CDS. A meaningful optimization problem therefore is to find a latency-s CDS of minimum cardinality. The notion of a latency-s CDS is relevant to s-club interdiction because of its close relationship to hereditary s-clubs as crystallized in the following result.

Proposition 4. *Consider a graph $G = (V, E)$ in which S is an s-club and $H \in \mathcal{H}(S)$ such that $H \neq S$. Then $S \setminus H$ is a latency-s CDS over S. Conversely, suppose that a non-empty $D \subseteq S$ is a latency-s CDS over S. Then $S \setminus D \in \mathcal{H}(S)$.*

Proof. ($\Longrightarrow$) The claim is trivial if H is empty; suppose not. Because S and $S \setminus H$ are s-clubs, it follows that $G[S]$ and $G[S \setminus H]$ are both connected. Hence, $S \setminus H$ dominates $G[S]$. It suffices to show that between distinct, non-adjacent vertices u and v in S, there exists a path of length at most s whose internal vertices belong to $S \setminus H$. The claim is trivially true if u and v belong to $S \setminus H$.

Suppose u and v belong to H. Define $T := H \setminus \{u, v\}$. Because S is an H-hereditary s-club, we know that $S \setminus T$ is an s-club that contains u and v. Hence, there exists a path of length at most s between u and v in $G[S \setminus T]$, and the internal vertices on this path clearly do not intersect H.

Now suppose $u \in S \setminus H$ and $v \in H$. Define $T' := H \setminus \{v\}$. As before, $S \setminus T'$ is an s-club that contains both u and v, and the internal vertices on some path of length at most s between them in $G[S \setminus T]$ are all contained in $S \setminus H$.

($\Longleftarrow$) For an arbitrary $T \subseteq S \setminus D$, we need to show that $S \setminus T$ is an s-club. Consider distinct, non-adjacent vertices u and v in $S \setminus T$. By definition, there exists a u, v-path of length at most s in $G[S]$ with all its internal vertices contained in D. None of these vertices are deleted when T is deleted, and the path is preserved in $G[S \setminus T]$. $\square$

Proposition 4 allows us to find large subsets $H \in \mathcal{H}(S)$ by equivalently finding small latency-s CDSs. Hence, when identifying violated constraints in our delayed constraint generation approach, we can replace the problem of finding a large $H \in \mathcal{H}(S)$ by finding a minimum cardinality latency-s CDS sets in S. By framing the problem in this manner we can exploit existing methods to solve the minimum latency-s CDS problem (Validi and Buchanan, 2020).

In Chapter IV, we use these results to design a decomposition branch-and-cut algorithm to solve Formulation (3.14), and perform numerical experiments to evaluate the capabilities of the proposed algorithm to solve the s-club interdiction problem on benchmark instances.

3.7 Interdicting cores

In this section, we study the problem of interdicting k-core of an undirected graph. A subset of vertices K is called a k-core of G, if the minimum degree of $G[K]$ is at least k, and K is maximal with respect to inclusion (Seidman, 1983). For convenience, we also refer to $G[K]$ as a k-core. A k-core is a useful model to capture cohesive subgroups and measure user engagement in social media assuming that the engagement depends on the number of connections a user has on the platform (Zhang et al., 2017). Matula and Beck (1983) provide a linear-time algorithm for computing the k-core for every value of k. A very fast implementation of this algorithm appears in Walteros and Buchanan (2020).

While working on this problem, we became aware of the existence of related papers in the literature (Zhang et al., 2017; Cerulli et al., 2022) that partially addressed the research questions for k-cores of interest to us. As a consequence, we decided to not pursue this model further. In the following sections, we briefly describe the directions we explored in our preliminary studies.

3.7.1 An MILP formulation

Similar to club interdiction, the k-core interdiction problem with an interdiction penalty can be formulated as a mixed integer linear programming as follows:

$$z_{s,\alpha} = \min\ \theta + \alpha x(V) \tag{3.37a}$$

$$\text{s.t.}\quad \theta \geq |K| - |K|x(K) \qquad \forall K \in \mathcal{K} \tag{3.37b}$$

$$x \in \{0,1\}^{|V|}, \theta \in \mathbb{R}_+, \tag{3.37c}$$

where θ is the cardinality of a largest k-core in the interdicted graph, and $\mathcal{K}$ is the set of all the k-cores in G. In general, k-cores are not hereditary and the Constraint (3.37b) associated with a k-core K becomes redundant when a vertex $v \in K$ is deleted. Therefore, we investigate the possibility of rewriting this constraint using the concepts of heredity and criticality that were introduced for s-clubs in Section 3.4. In this regard, we have the following:

Definition 9. *Given a graph $G = (V, E)$, a k-core K in G, and $H \subseteq K$, we say that K is an H-hereditary k-core if $\delta_{K\setminus T} \geq k$ where $\delta_{K\setminus T} = \min\{\deg_{K\setminus T}(v) : v \in K \setminus T\}$ for every $T \subseteq H$.*

We denote by $\mathcal{H}(K)$ the set of all the hereditary subsets of K. Also, we define $\mathcal{U}(K)$ in an analogous way $\mathcal{U}(S)$ is defined in Equation (3.10).

Definition 10. *The set of critical k-cores in the graph $G = (V, E)$ is defined as $\mathcal{C}^* = \{K \in \mathcal{K} \mid$ no k-core $K' \supset K$ exists such that $K \in \mathcal{U}(K')\}$.*

Proposition 5. *Each k-core K in graph $G = (V, E)$ is critical.*

Proof. For the sake of contradiction, suppose there exists a k-core K in G that is not critical. Therefore, we can write it as $K = K^0 \setminus T$ where K^0 is a k-core and $T \subseteq \mathcal{H}(K^0)$. This means that we can write $K^0 = K \cup T$, so K is not maximal. This is a contradiction because a k-core is a maximal connected subgraph with a minimum degree of at least k for every vertex in K.

So, K has to be critical. Since K is chosen arbitrarily, we can conclude that any k-core in G is critical. $\square$

Proposition 5 shows that the techniques we have used for the s-club interdiction problem to improve the formulation by considering only critical s-clubs and reducing the number of constraints will not be beneficial for the k-cores. Therefore, in the next sections, we examine other approaches to solve the problem.

3.7.2 Testing for matroid and submodularity properties

In this section, we investigate if the maximum k-core in a graph post interdiction is a submodular function and if the family of hereditary sets of a k-core induces a matroid. The motivation for testing these properties is that these types of functions/systems have properties akin to convex functions and play an important role in combinatorial optimization (Feige et al., 2011). For example, it is proved that under some conditions, there is a polynomial time algorithm for minimization of any submodular function (Grötschel et al., 1981). Likewise, some matroid optimization problems can be solved in polynomial time using greedy algorithms (Brezovec et al., 1986). Therefore, testing these properties can be useful to find potentially fast algorithms to solve the k-core interdiction problem.

First, we consider $M = (K, \mathcal{H}(K))$ where K is a k-core in G and $\mathcal{H}(K)$ is the set of all of the hereditary subsets of K based on Definition 9. M should have the following properties to be a matroid (Welsh, 2010):

1. $\emptyset \in \mathcal{H}(K)$.
2. For each $A' \subseteq A \subseteq K$, if $A \in \mathcal{H}(K)$, then $A' \in \mathcal{H}(K)$.
3. Let $A \in \mathcal{H}(K)$, $B \in \mathcal{H}(K)$, and $|A| > |B|$. Then, there exist $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{H}(K)$.

We see that M in an independence system, because it has the first two properties: (1)$\emptyset$ is a hereditary subset for every k-core, (2) based on the definition of heredity, if $H \subseteq K$ is a hereditary subset, then $T \in \mathcal{H}(K)$ for every $T \subseteq H$. This shows that M is an independent system. However, M is not a matroid as it does not have the third property. Figure 3.5 shows a counterexample.

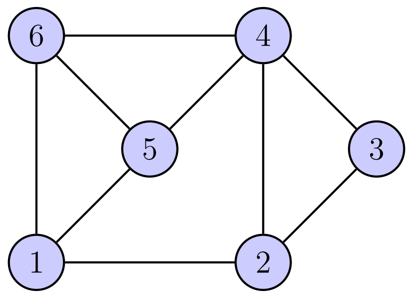

Figure 3.5: $K = \{1, ..., 6\}$ is a 2-core and $\mathcal{H}(K) = \big\{\emptyset, \{1\}, \{3\}, \{5\}, \{6\}, \{3, 5\}, \{3, 6\}\big\}$. Let $A = \{3, 6\}$ and $B = \{1\}$. Then, $A \setminus B = A$, but none of the sets $B \cup \{3\}$ and $B \cup \{6\}$ are in $\mathcal{H}(K)$.

Next, we define $\phi(S)$ as the k-core number of $G \setminus S$, i.e., the size of the largest k-core in G after S is interdicted. Using a counterexample, we can show this function is neither submodular nor supermodular. Inequalities (3.38) and (3.39), respectively, show the condition for submodularity and supermodularity of $\phi(S)$ (Grötschel et al., 1981):

$$\phi(X \cup \{x\}) - \phi(X) \geq \phi(Y \cup \{x\}) - \phi(Y) \qquad \forall\, X, Y \subseteq V, X \subseteq Y, x \in V \setminus Y \tag{3.38}$$

$$\phi(X \cup \{x\}) - \phi(X) \leq \phi(Y \cup \{x\}) - \phi(Y) \qquad \forall\, X, Y \subseteq V, X \subseteq Y, x \in V \setminus Y \tag{3.39}$$

Consider the counterexample in Figure 3.6. If $k = 2, X = \{2\}$, $Y = \{2, 3\}$ and $x = \{1\}$, we have $\phi(X) = 4$, $\phi(Y) = 3$, $\phi(X \cup \{x\}) = \phi(Y \cup \{x\}) = 0$. As it can be seen the inequality (3.38) does not hold, so $\phi(S)$ is not submodular. Also, if $x = \{5\}$, we have $\phi(X) = 4$, $\phi(Y) = 3$, $\phi(X \cup \{x\}) = 3$, and $\phi(Y \cup \{x\}) = 0$. Therefore, the inequality (3.39) does not hold, and $\phi(S)$ is not supermodular.

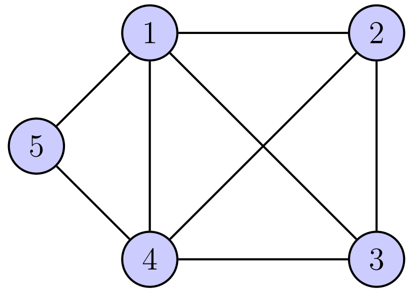

Figure 3.6: Counterexample to show $\phi(S)$ and $\Omega(S)$ are neither submodular nor supermodular.

Now consider $\Omega(S) = |V| - \phi(S)$. We show that $\Omega(S)$ is not submodular using the counterexample in Figure 3.6. Let $k = 2, X = \{2\}$, $Y = \{2, 3\}$ and $x = \{5\}$. Then we have $\Omega(X) = 1$, $\Omega(Y) = 2$, $\Omega(X \cup \{x\}) = 2$ and $\Omega(Y \cup \{x\}) = 5$. It can be seen that inequality (3.38) does not hold because $2 - 1 \not\geq 5 - 2$, and $\Omega(S)$ is not submodular. This function is not supermodular either because if $x = \{1\}$, we have $\Omega(X) = 1$, $\Omega(Y) = 2$, $\Omega(X \cup \{x\}) = 5$ and $\Omega(Y \cup \{x\}) = 5$. Therefore, the inequality (3.39) does not hold because $5 - 1 \not\leq 5 - 2$, and $\Omega(S)$ is not supermodular.

Based on these observations, we can conclude the algorithms for optimizing submodular functions and for optimizing over matroids cannot be used in our setting, at least in a straightforward manner. In the next section, we study the k-core interdiction problem with a budget constraint instead of the interdiction penalty.

3.7.3 Budgeted version of the problem

Zhang et al. (2017) proposed the collapsed k-core problem defined as finding the set of b vertices in $G = (V, E)$ whose deletion results in the smallest k-core. They prove that the problem is NP-hard and propose a greedy heuristic algorithm to find the interdiction set with a limited budget. The complexity of the heuristic algorithm is $O(bnm)$ where b, n, and m are respectively the budget, number of nodes, and number of edges in the graph. At each iteration, the algorithm removes a vertex whose deletion results in the smallest k-core in the remaining graph and the number of iterations is equal to the budget. Figures 3.7-3.10 show an example of an instance where the solution of the algorithm is not optimal.

Consider the original graph in Figure 3.7 which is also a 3-core and suppose $b = 2$. According to the proposed algorithm by Zhang et al. (2017), in the first iteration, vertex 7 must be deleted. The remaining 3-core is shown in Figure 3.8. In the second iteration, deleting any of the vertices will result in a 3-core of size 4 (see Figure 3.9). Therefore, with 2 units of budget, the solution of the algorithm is 4. However, if we delete vertices 5 and 9, the size of the 3-core will be zero and this is an optimal solution, see Figure 3.10.

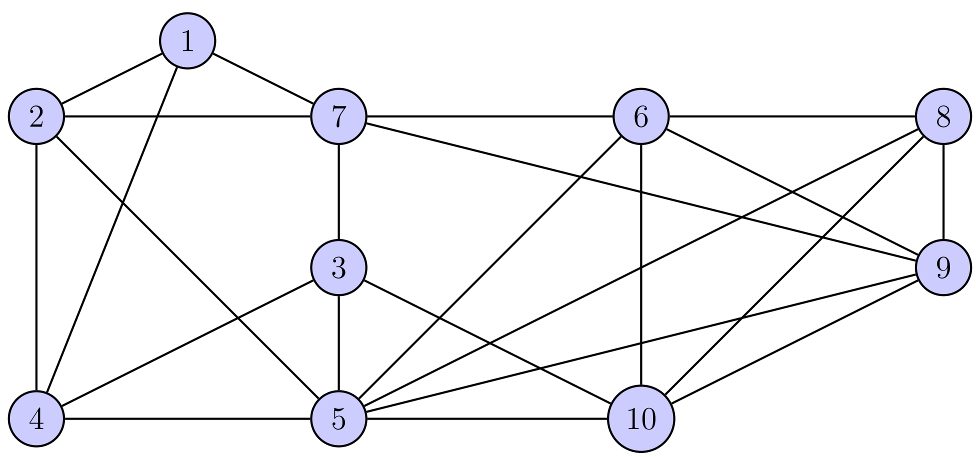

Figure 3.7: Graph G that is a 3-core.

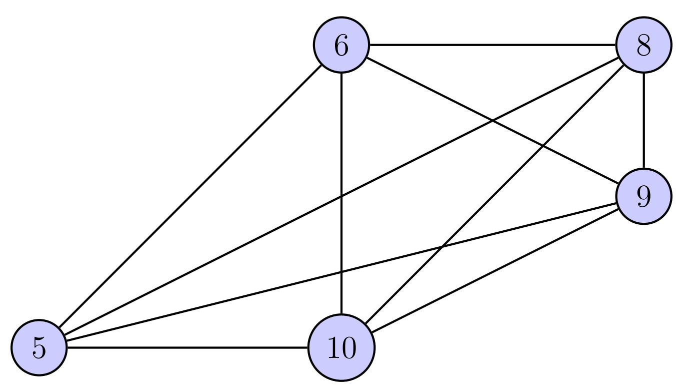

Figure 3.8: The 3-core obtained by removing vertex 7 from graph G in Figure 3.7

Alternatively, consider the following. By interdicting every vertex in a k-core, the degree

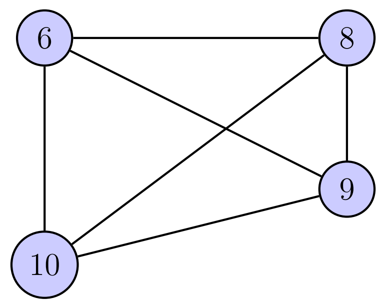

Figure 3.9: Removing any of the vertices from the graph in Figure 3.8 will result in a 3-core of size 4. Here, the 3-core by deleting vertex 5 is shown.

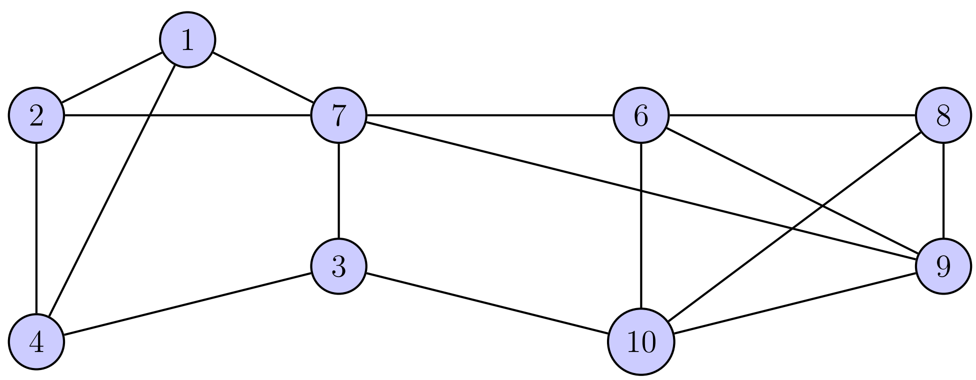

Figure 3.10: Deleting vertex 5 from the graph in Figure 3.7 results in a 3-core with size 9, so the algorithm proposed by Zhang et al. (2017) does not pick this vertex in the first iteration. However, by deleting both vertices 5 and 9, the size of the 3-core will be zero.

of its neighbors decreases by at least 1. This implies that in addition to the interdicted vertex, all the neighbors with degree equal to k will be removed as well because they do not have the minimum degree to be in the k-core. We use this fact to propose a greedy algorithm to find the interdiction set. Let K be the largest k-core in the graph. At each iteration, this algorithm determines a set L containing the vertices in K with the smallest degree. It also finds the size of the set $L \cap N_K(v)$ for every vertex $v \in K$. Then, the vertex with the largest intersection will be deleted. This choice can be justified as follows: if the minimum degree is k, deleting this vertex might result in removing more vertices in the same iteration; if the minimum degree is greater than k, deleting this vertex decreases the degree of more vertices in set L and make their degree closer to k, so potentially, more vertices will be removed in the next iterations. It is readily checked that this algorithm (see Algorithm 1) runs in polynomial time. However, it is not guaranteed to find an optimal solution.

Algorithm 1: A greedy algorithm for the maximum k-core interdiction problem.

Input: $G(V, E)$, integer k, budget b
Output: Largest k-core in $G \setminus D$ where D is the deletion set and $|D| \leq b$.
$K \leftarrow$ maximum k-core in G
while $b > 0$ **do**
  $G' \leftarrow G[K]$
  $\delta(G') \leftarrow \min\{\deg_{G'}(v) : v \in V\}$
  $L \leftarrow \{u \in K : \deg_{G'}(u) = \delta(G')\}$
  $D \leftarrow D \cup \{v \in \operatorname{argmax}_{u \in K} |L \cap N_{G'}(u)|\}$
  $b \leftarrow b - 1$
  $K \leftarrow$ maximum k-core $(G' \setminus D)$
Return K

Figure 3.11 shows an example where this algorithm does not find an optimal interdiction solution. Let $k = 2$ and $b = 1$. In this 2-core, we have $L = \{2, 3, 5, 6\}$ and the size of the intersection of L with the neighbors of vertices is equal to two for vertices 1, 4 and 6, is equal to one for vertices 2 and 4 and is zero for vertex 5. Therefore, vertices 1, 4, and 6 are all candidates to be deleted. However, $K_{G\setminus\{1\}} = K_{G\setminus\{4\}} = 0$ and $K_{G\setminus\{6\}} = 3$, so the solution $D = \{6\}$ is not optimal.

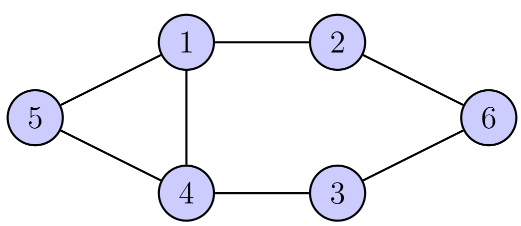

Figure 3.11: Using Algorithm 1, vertices 1, 4, and 6 are all candidates to be deleted. However, $D = \{6\}$ is not optimal because $K_{G\setminus\{1\}} = K_{G\setminus\{4\}} = 0$ and $K_{G\setminus\{6\}} = 3$.

Previous research on this topic includes the NP-hardness of the problem, fixed parameter tractability, and $W[1]$-hardness (Luo et al., 2021). More recently, an integer programming approach for the collapsed k-core problem has been introduced by Cerulli et al. (2022). They provide two bilevel programs for the collapsed k-core problem. One of the programs is reformulated as a single-level model that exploits a Benders-like decomposition approach. For the other program, the lower-level problem is stated as a linear formulation to find the k-core. They also derive combinatorial lower bound on the value of the optimal solution, and describe some pre-processing techniques and valid inequalities for all the formulations they have proposed.

CHAPTER IV

COMPUTATIONAL EXPERIMENTS WITH MAXIMUM S-CLUB INTERDICTION

In this chapter, we present a decomposition algorithm to implement the formulations for the maximum s-club problem. Also, we discuss the numerical experiments in detail and evaluate the effect of reformulating the problem using the critical s-clubs and their hereditary subsets.

4.1 Implementing a decomposition branch-and-cut algorithm

Based on the results of Sections 3.4 and 3.6, our approach to solve Formulation (3.14) employs delayed constraint generation in a decomposition and branch-and-cut (DBC) framework. This DBC algorithm starts by solving an initial relaxation of Formulation (3.14) where $\mathcal{C}^*$ in constraint (3.14b) is replaced by an initial collection of s-clubs $\mathcal{S}^0 \subseteq \mathcal{S}$. As this initial relaxation is solved using an LP relaxation based branch-and-cut (BC) algorithm, nodes are pruned as usual by infeasibility or by bound. However, if the LP relaxation optimum (θ^i, x^i) at some BC node i is integral, we must verify its feasibility.

To this end, we can solve a separation subproblem in order to verify if a constraint of type (3.14b) corresponding to some H-hereditary s-club S is violated. First, we find a maximum s-club in the interdicted graph, say S. If $\bar{\omega}_s(G \setminus T^{x^i}) = |S| > \theta^i$, we must add a violated constraint to eliminate this solution. In order to find an $H \in \mathcal{H}(S)$, based on Proposition 4, we can solve the minimum latency-s CDS problem on the subgraph $G[S]$. If $\mathcal{H}(S)$ is empty, then the minimum latency-s CDS will be S itself, and we add constraint (3.3b)

Algorithm 2: Separation procedure

Input: Integral LP optimum $(\hat{\theta}, \hat{x})$ at the DBC node.
Output: (S, H), where S is an H-hereditary s-club corresponding to a violated constraint, if one exists.

$S \leftarrow$ a maximum s-club in $G \setminus T^{\hat{x}}$
if $|S| > \hat{\theta}$ **then**
  $D \leftarrow$ a minimum latency-s CDS in $G[S]$
  return $(S, S \setminus D)$
else
  $(\hat{\theta}, \hat{x})$ is feasible

for S. After the violated constraint is added, the LP relaxation at node i is re-solved. If $\bar{\omega}_s(G \setminus T^{x^i}) = |S| \leq \theta^i$, no violated constraint exists, we can certify that the integral solution (θ^i, x^i) is feasible to the original problem and prune that BC node. This separation routine is described in Algorithm 2.

The separation subproblem ensures the correctness of the overall algorithm despite starting with a relaxation of the original problem. From our experiments, we found that the DBC algorithm typically generates far fewer constraints than all possible constraints of type (3.14b). In the following we discuss how we initialize the relaxation problem in our computational study described in Section 4.2, as well as specify some additional implementation details of the heuristic separation procedure used in our experiments when $s = 2$ and $s = 3$.

4.1.1 Implementation details for 2-club interdiction

When solving the 2-club interdiction problem, we initialize the relaxation problem with constraints based on stars in G (see Remark 2). We write the constraints for the star $N_G[v]$ centered at v, with the hereditary subset $H = N_G(v)$. This choice of H is maximal as long as $N_G[v]$ is not a clique. Hence, the initial relaxation constraints have the following form:

$$\theta \geq \deg_G(v) + 1 - x(N_G(v)) - (\deg_G(v) + 1)x_v. \tag{4.1}$$

In our experiments, we add constraint (4.1) only for those vertices that correspond to the top 20% of the largest degrees in G to consider larger 2-clubs (in form of stars) and avoid adding too many constraints in advance.

Once a maximum 2-club S that corresponds to a violated constraint is found, we use a simpler heuristic approach to identify a hereditary subset for the case of $s = 2$, instead of finding a minimum latency-s CDS inside $G[S]$ (line 3 of Algorithm 2). This simplification is based on the observation that if $G[S]$ contains a dominating vertex v, then the set $\{v\}$ is a latency-2 CDS and S is a $S \setminus \{v\}$-hereditary 2-club. In fact, if S is not a clique, then $\{v\}$ is a *minimum* latency-2 CDS of $G[S]$.

Algorithm 3 outlines the pseudocode of a heuristic separation procedure for $s = 2$ that does not rely on solving the minimum latency-s CDS problem. If we find any vertex v that dominates $G[S]$, we return immediately having identified a strong violated constraint. Otherwise, we find all the leaves L in $G[S]$ and $S \setminus L$ is a feasible latency-2 CDS. If no leaves exists, then L is empty, and we effectively add a constraint of type (3.3b). In all three cases, note that the constraint identified will be violated by $(\hat{\theta}, \hat{x})$. This heuristic separation procedure was found to be effective for the case $s = 2$ in our computational studies.

4.1.2 Implementation details for 3-club interdiction

In this case, the initial set $\mathcal{S}^0$ of 3-clubs includes the constraints associated with edge stars corresponding to $S \coloneqq N_G(u) \cup N_G(v)$ for each $\{u, v\} \in E$, with $H = N_G(u) \cup N_G(v) \setminus \{u, v\}$. Clearly, $H \in \mathcal{H}(S)$, and H would belong to $\mathcal{H}^*(S)$ unless $H \cup \{u\} \in \mathcal{H}^*(S)$, $H \cup \{v\} \in \mathcal{H}^*(S)$, or $H \cup \{u, v\} \in \mathcal{H}^*(S)$. In other words, H is at most two elements short of a maximal member of $\mathcal{H}(S)$ in case it is not in $\mathcal{H}^*(S)$. The constraint of type (3.14b) specializes to the following for edge stars:

$$\theta \geq |S| - x(S \setminus \{u, v\}) - |S|(x_u + x_v) \quad \forall \{u, v\} \in E. \tag{4.2}$$

Algorithm 3: Separation algorithm for $s = 2$

Input: Integral LP optimum $(\hat{\theta}, \hat{x})$ at the DBC node.
Output: (S, H), where S is a H-hereditary 2-club corresponding to a violated constraint, if one exists.

$S \leftarrow$ a maximum 2-club in $G \setminus T^{\hat{x}}$
if $|S| > \hat{\theta}$ **then**
  $L \leftarrow \emptyset$
  for $v \in S$ **do**
    if $|N_{G[S]}(v)| = |S| - 1$ **then**
      return $(S, S \setminus \{v\})$
    if $|N_{G[S]}(v)| = 1$ **then**
      $L \leftarrow L \cup \{v\}$
  return (S, L)
else
  $(\hat{\theta}, \hat{x})$ is feasible

As every 2-club is also a 3-club, we also add constraint (4.1) for all the vertices in G. In general, the star constraints are not dominated by edge star constraints (4.2).

4.2 Computational experiments

In this section, we report on the results of our numerical experiments designed to assess the capabilities of the proposed DBC algorithm to solve the s-club interdiction problem on real and synthetic benchmark instances. All experiments are conducted on a 64-bit Windows® 10 Pro machine with 16GB of RAM and 1.8 GHz processor with 7 cores. All algorithms are implemented in C++, compiled using Microsoft® Visual Studio® 2017, and Gurobi™ Optimizer v9.0.2 is used to solve the MILPs (Gurobi Optimization, LLC, 2021). Our codes are publicly available (Daemi et al., 2021b,a).

Our testbed consists of two groups of instances. Group-1 contains 22 graphs from the Tenth DIMACS Implementation Challenge (DIMACS-10), see (Dimacs, 2012). Group-2 contains 18 graphs taken from the following online repositories: Stanford Large Network Dataset

Collection (SNAP) (Leskovec and Krevl, 2014), the BGU Social Networks Security Research Group (BGU) (Lesser et al., 2013), the Koblenz Network Collection (KONECT) (Kunegis, 2013) and the Network Repository (NR) (Rossi and Ahmed, 2015). Most of the instances in our testbed come from real-world networks. Further, the instances `Gplus, Facebook1, Facebook2,` and `Douban` in Group-2 represent snapshots of real online social networks. The instances in Group-2 were also used in the computational studies in Raghavan and Zhang (2019).

Tables 4.1 and 4.2 list all the graphs in our testbed. We converted the directed graphs to undirected graphs by ignoring the orientation on the arcs. For each instance we list the number of vertices, edges, and the edge density $\rho(G) = |E|/\binom{|V|}{2}$. To solve the maximum s-club problem during separation, we use the "ICUT" algorithm introduced by Salemi and Buchanan (2020), the code for which has been made publicly available by the authors. ICUT is an effective integer-programming-based exact solver for the maximum s-club problem for general values of s on the instances we use in our testbed. It sequentially solves the maximum s-club problem on several smaller subgraphs using a delayed constraint generation framework. Tables 4.1 and 4.2 report the time it takes to find $\bar{\omega}_2(G)$ and $\bar{\omega}_3(G)$ using the ICUT solver. TL in the Time column indicates that the solver terminated by reaching the time limit.

Using the Gurobi parameter `GRB_DoubleParam_Timelimit`, we impose a time limit of 3600 seconds on the solve time of the initial problem, and the same time limit on each call to solve the maximum s-club subproblems in ICUT and the minimum latency-s CDS problem during the separation procedure. Reaching the time limit while solving any of these problems will terminate the overall algorithm (usually quickly), in which case we have failed to solve the problem to optimality on that instance. We also use the Gurobi parameter `LazyConstraint` to add the violated constraints found in the separation procedure on-the-fly.

As discussed in Section 4.1, the DBC algorithm requires solving the maximum s-club problem several times, once for every integral solution $(\hat{\theta}, \hat{x})$ that is found in the BC tree to

Table 4.1: DIMACS-10 instances in Group-1 and the time taken to solve the maximum s-club problem for $s = 2, 3$ using the ICUT algorithm. Instances `celegansneural`, `celegans-metabolic`, and `PGPgiantcompo` are shortened to `celegansn`, `celegansm`, and `PGP`, respectively, in the other tables.

Graph G	$\|V\|$	$\|E\|$	$\rho(G)$ (%)	$\bar{\omega}_2(G)$	Time (s)	$\bar{\omega}_3(G)$	Time (s)
karate	34	78	13.90	18	0.01	25	0.00
dolphins	62	159	8.41	13	0.14	29	0.02
lesmis	77	254	8.68	37	0.00	58	0.00
polbooks	105	441	8.08	28	0.09	53	0.00
adjnoun	112	425	6.84	50	0.00	82	0.19
football	115	613	9.35	16	0.84	58	1.52
jazz	198	2,742	14.06	103	0.42	174	0.05
celegansneural	297	2,148	4.89	135	0.02	243	0.37
celegans-metabolic	453	2,025	1.98	238	0.02	371	0.10
email	1,133	5,451	0.85	72	6.89	212	65.69
polblogs	1,490	16,715	1.51	352	30.82	776	31.43
netscience	1,589	2,742	0.22	35	0.02	54	0.02
add20	2,395	7,462	0.26	124	0.17	671	0.23
data	2,851	15,093	0.37	18	13.27	32	15.51
uk	4,824	6,837	0.06	5	12.32	8	13.86
power	4,941	6,593	0.05	20	0.68	30	0.69
add32	4,960	9,462	0.08	32	0.48	99	0.50
hep-th	8,361	15,751	0.05	51	1.34	120	41.66
whitaker3	9,800	28,989	0.06	9	66.50	15	90.78
crack	10,240	30,380	0.06	10	81.95	17	96.06
PGPgiantcompo	10,680	24,316	0.04	206	4.07	422	4.30
cs4	22,499	43,858	0.02	6	165.26	12	236.51

Table 4.2: Instances in Group-2 and the time taken to solve the maximum s-club problem for $s = 2, 3$ using the ICUT algorithm.

Graph G	Source	$\|V\|$	$\|E\|$	$\rho(G)$(%)	$\bar{\omega}_2(G)$	Time (s)	$\bar{\omega}_3(G)$	Time (s)
G04	SNAP	10,876	39,994	0.07	104	4.89	≥ 181	TL
G05	SNAP	8,846	31,839	0.08	89	9.96	≥ 258	TL
G06	SNAP	8,717	31,525	0.08	116	3.64	≥ 243	TL
G08	SNAP	6,301	20,777	0.10	98	23.08	453	464.72
G09	SNAP	8,114	26,013	0.08	103	20.93	449	945.33
B-Alpha	SNAP	3,783	14,124	0.20	512	0.66	1,294	626.06
B-OTC	SNAP	5,881	21,492	0.12	796	1.36	$\geq$ 1,969	TL
AS01	SNAP	10,670	22,002	0.04	2,313	15.25	4,997	613.26
AS02	SNAP	10,900	31,180	0.05	2,344	15.89	5,352	202.68
Ning	BGU	10,298	40,887	0.09	688	4.29	$\geq$ 2,294	TL
Hamsterster	Konect	1,858	12,534	0.78	273	0.18	680	89.18
Escorts	Konect	10,106	39,016	0.08	312	4.32	≥ 679	TL
Anybeat	N.R.	12,645	49,132	0.06	4,801	9.17	$\geq$ 7,752	TL
Advogato	N.R.	6,551	39,432	0.31	808	1.64	2,193	1,937.74
Gplus	Konect	23,613	39,194	0.01	2,762	9.99	$\geq$ 4,767	TL
Facebook1	BGU	39,446	50,228	0.01	1,366	27.45	11,542	2,136.21
Facebook2	Konect	2,888	2,981	0.07	770	0.13	1,241	0.18
Douban	N.R.	154,908	327,162	0.00	≥ 288	TL	≥ 911	TL

verify its feasibility. Therefore, if solving the maximum s-club problem requires a significant amount of time for a given graph, then we do not expect the interdiction problem to be solved in a reasonable amount of time. More critically, reaching the time limit on the maximum s-club solver without producing a violated s-club affects the overall correctness. For this reason, we only consider those instances in the larger test bed described next on which we can find a large enough s-club in reasonable time using our chosen solver.

As it can be seen in Tables 4.1 and 4.2, all instances in Group-1 are solved within a reasonable time (less than 5 minutes) for both $s = 2$ and $s = 3$. For Group-2, all the instances except `Douban` are solved to optimality when $s = 2$. However, when $s = 3$, only 9 out of 18 instances are solved to optimality within the time limit, and among these instances, `Advogato` and `Facebook1` requires a significant amount of time. For this reason, when $s = 2$, we do not include instance `Douban` in our experiments and when $s = 3$, for instances in Group-2, we use heuristic approaches to find a maximum 3-club and a minimum latency-3 CDS instead of using the exact methods we implement in other cases.

In Section 4.2.1, we use the Group-1 instances to show how the naive Formulation (3.3) and Formulation (3.14) based on hereditary s-clubs compare when each is used in the DBC algorithm. In Sections 4.2.2 and 4.2.3, we present the results of our experiments with both groups of instances using the best performing DBC algorithm and heuristic approaches.

4.2.1 The impact of using the H-hereditary s-club formulation

In Section 3.5, we introduced the idea of partitioning an s-club S into a hereditary subset H and $S \setminus H$ in order to generate a constraint of type (3.14b). Here, we assess the impact of using these constraints by comparing three different methods. In the first method, a constraint of type (3.3b) is used in the initialization and during separation (Method 1). The second method uses the H-hereditary s-club constraint (3.14b) in the initialization of the relaxation problem and constraint (3.3b) during separation (Method 2). The third method uses constraint (3.14b) during initialization and separation (Method 3). In all three methods, we initialize $\mathcal{S}^0$ by creating a set of s-clubs in the form of stars (when $s = 2, 3$) or edge stars (when $s = 3$), and add a constraint for each s-club in $\mathcal{S}^0$ to initialize the relaxation problem. Note that the type of constraint we add for each s-club in $\mathcal{S}^0$ depends on the method, as explained before. We compare the performance of these three methods in terms of running time and visualize the comparison using performance profiles (Dolan and Moré, 2002).

In order to construct a performance profile, we define $\mathcal{I}$ as the set of the instances in our testbed, $\mathcal{M}$ as the set of methods, and $t_{i,m}$ as the running time of solving the instance i by method m. The baseline of the comparison is the shortest running time among three methods for every instance, and we compute the performance ratio as $r_{i,m} = t_{i,m}/t_i^*$, where $t_i^* = \min\{t_{i,m} : m \in \mathcal{M}\}$. Then, for every method m, we define $f_m(\tau)$ as the empirical cumulative distribution function of the performance ratio $r_{i,m}$. As stated in Equation (4.3), $f_m(\tau)$ is the fraction of the instances in our testbed that were solved by method m within a factor τ of the fastest solve-time for that instance.

$$f_m(\tau) = \frac{|\{i \in \mathcal{I} : t_{i,m} \leq \tau t_i^*\}|}{|\mathcal{I}|}. \tag{4.3}$$

If we observe that $f_m(\tau) \geq f_{m'}(\tau)$ for all $\tau \geq 1$, then there is evidence to suggest that method m is better than m' on this testbed. In particular, $f_m(1.0)$ is the fraction of the instances in the testbed for which method m is the fastest. It is best to interpret these profiles as the comparison of Method 3 vs Method 1 and Method 3 vs Method 2 for each value of $\alpha \in \{0.5, 2\}$ Gould and Scott (2016).

Figure 4.1 shows the performance profiles of Method 1, Method 2, and Method 3 for all the instances in Group-1 for $s = 2$. We selected $\alpha = 0.5$ and $\alpha = 2$ for these experiments, meaning that in the former setting it is cheap to interdict vertices (i.e., for every two interdicted vertices the maximum s-club in the interdicted graph should reduce by at least one) while in

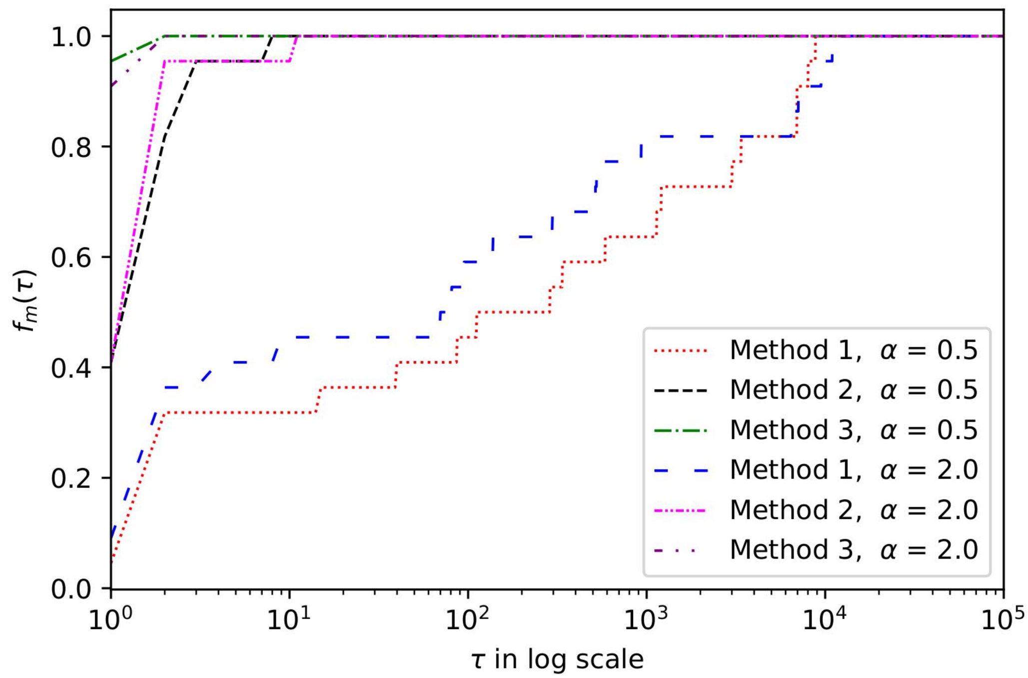

Figure 4.1: Performance profile based on the running time of methods for $s = 2$ and $\alpha \in \{0.5, 2\}$.

the latter setting it is expensive to interdict vertices (i.e., for every interdicted vertex the maximum s-club in the interdicted graph should reduce by at least two).

It can be seen that for both values of α, Method 3 is significantly better than Methods 1 and 2 on this testbed for $s = 2$. The performance of Method 2 is generally within 10 times the fastest running time, while Method 1 has a far worse performance overall, achieving 10 times the fastest running time only for less than 50% of the instances.

The performance profile for $s = 3$ is shown in Figure 4.2. As before, Method 3 has the best performance on this testbed. Method 2 performs worse than it did when $s = 2$, because there are about 5% of the instances whose solution times are not within 100 times the fastest running time when $\alpha = 2$. Method 1, on the other hand, has a similarly poor performance now, as it was the case with $s = 2$.

These comparisons show that, in general, Method 3 outperforms the other two methods.

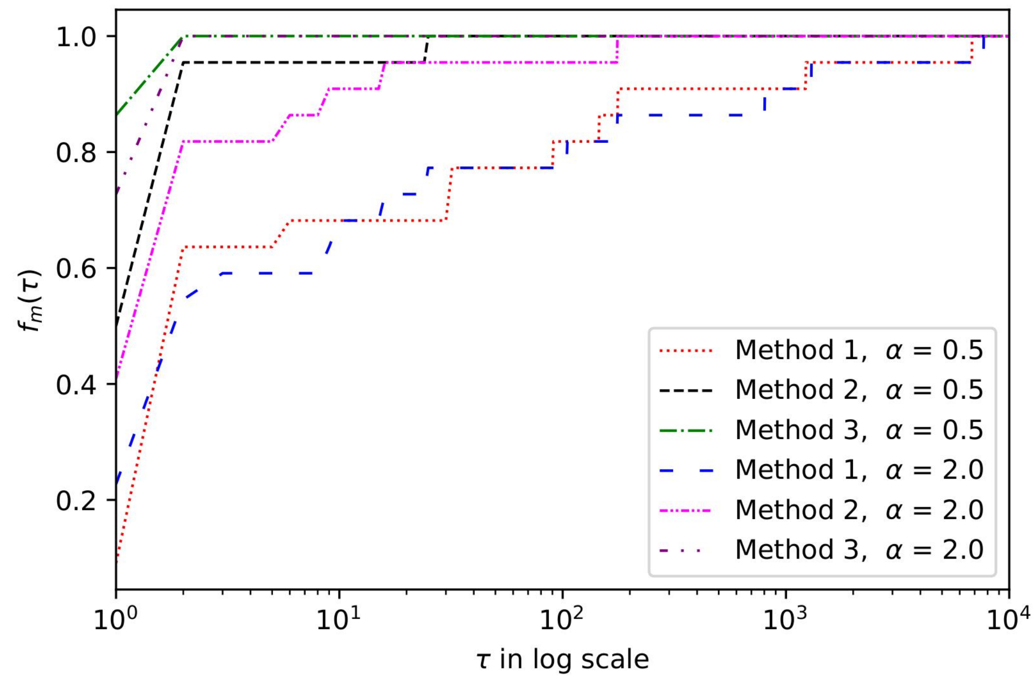

Figure 4.2: Performance profile based on the running time of methods for $s = 3$ and $\alpha \in \{0.5, 2\}$.

This observation confirms that using constraints based on H-hereditary s-clubs at initialization and during separation can significantly improve the performance of our DBC algorithm. Therefore, we use this method in the remaining computational experiments in Sections 4.2.2 and 4.2.3.

Before discussing the results of our main experiments with Method 3, we should mention that we evaluated its performance by conducting two other experiments reported in greater detail in Section 4.3. First, a root node performance comparison between Method 1 and Method 3. The results show that Method 3 outperforms Method 1 by providing the same or smaller gaps and objective values for nearly all the instances (see Section 4.3.1). We also evaluated the dependency of Method 3 on primal heuristics built into the Gurobi solver, comparing its performance with and without these heuristics. Neither choice consistently offers superior performance, and we discuss this in greater detail in Section 4.3.2.

4.2.2 Results for Group-1 instances

We report on the results obtained for the instances in Group-1 for $s \in \{2, 3\}$ and $\alpha \in \{0.5, 1, 2\}$ using Method 3 in this section. For each instance, we report the number of interdicted vertices under $x(V)$, the s-club number of the interdicted graph under θ, the total number of BC nodes explored, the total number of separation callbacks under #CB, the total number of violated constraints added under #Cuts (broken down by each type when $s = 2$ under Star, Leaf, Regular), the total running time, the total time taken to solve the maximum s-club problem, the total time taken to solve the minimum latency-s CDS problem (when $s = 3$), and the relative optimality gap at termination.

Tables 4.3, 4.4, and 4.5 show the results for $\alpha = 2, 1$, and 0.5, respectively, with $s = 2$. All the instances are solved to optimality under a one hour time limit with the exception of `jazz` and `polblogs` that are not solved to optimality for any value of α. We can observe in Table 4.1 that for most of the instances the 2-club number of the original graph $\bar{\omega}_2(G)$ tends to be much larger than the 2-club number after interdiction (i.e., θ) for all values of α we consider. For example, the values of $\bar{\omega}_2(G)$ in the original graph for `celegans-metabolic` and `PGPgiantcompo` are respectively 238 and 206, while they decrease to 32 and 76 after interdiction when $\alpha = 2$. These values further decrease to 10 and 47 as $\alpha = 0.5$ because interdiction is cheaper in this case. However, when $\bar{\omega}_2(G)$ is very small compared to $|V|$, we find θ to be almost equal to $\bar{\omega}_2(G)$. Consider the instance `cs4` as an example, with $\bar{\omega}_2(G) = 6$. The 2-club number of this graph remains the same after interdiction for all values of α we considered (note that this instance has 22,449 vertices).

Another observation is that for most of the instances, decreasing the value of α from 2 to 0.5 makes the instance more difficult to solve and, as a result, the number of BC nodes explored and running times increase. For example, when $\alpha = 2$, `football` is solved in the root node in 3.07 seconds, while for $\alpha = 0.5$, the number of explored nodes is $973,384$ and the running time increases to 92.14 seconds. This behavior could be due to the fact that as α

Table 4.3: Results for Group-1 instances with $s = 2$ and $\alpha = 2$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Regular	Total time (s)	s-club time (s)	Gap (%)
karate	3	9	1	5	0	3	0.08	0.01	0.00
dolphins	0	13	1	2	1	0	0.10	0.06	0.00
lesmis	2	18	1	4	1	0	0.38	0.01	0.00
polbooks	1	25	1	14	4	8	0.70	0.45	0.00
adjnoun	6	14	1	5	1	0	0.56	0.46	0.00
football	0	16	1	12	0	10	3.07	3.01	0.00
jazz	5	71	37,284	4,735	1	4,731	TL	3570.47	16.17
celegansn	12	36	63	8	1	3	7.18	6.87	0.00
celegansm	13	32	21	4	1	0	0.33	0.03	0.00
email	1	52	1	3	1	0	12.37	12.11	0.00
polblogs	21	154	1,180	100	2	92	TL	3607.45	18.67
netscience	3	21	1	3	1	0	0.16	0.04	0.00
add20	14	68	32	5	1	0	6.88	0.81	0.00
data	0	18	0	2	0	1	7.64	7.58	0.00
uk	0	5	1	8	0	7	29.93	29.76	0.00
power	2	15	1	3	0	1	1.74	1.60	0.00
add32	0	32	1	2	1	0	0.76	0.63	0.00
hep-th	3	40	1	3	1	0	3.87	3.05	0.00
whitaker3	0	9	0	2	1	0	48.23	47.96	0.00
crack	0	10	1	2	1	0	38.09	37.86	0.00
PGP	11	76	79	6	4	0	26.46	20.02	0.00
cs4	0	6	1	6	0	5	407.60	405.63	0.00

Table 4.4: Results for Group-1 instances with $s = 2$ and $\alpha = 1$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Regular	Total time (s)	s-club time (s)	Gap (%)
karate	5	5	1	9	3	4	0.12	0.01	0.00
dolphins	0	13	34	4	2	1	0.27	0.12	0.00
lesmis	8	10	57	10	4	3	0.16	0.01	0.00
polbooks	13	12	144	12	6	3	0.42	0.14	0.00
adjnoun	6	14	23	6	1	0	1.01	0.80	0.00
football	0	16	1	14	0	11	3.52	3.42	0.00
jazz	21	45	74,282	6,389	22	6,361	TL	3598.92	10.42
celegansn	21	23	175	6	1	0	4.24	3.92	0.00
celegansm	21	18	54	5	1	0	0.35	0.05	0.00
email	8	42	49	3	1	0	12.72	11.38	0.00
polblogs	113	55	972	140	1	135	TL	3737.62	30.78
netscience	3	21	1	3	1	0	0.17	0.04	0.00
add20	30	49	281	5	2	0	8.98	0.64	0.00
data	0	18	1	2	0	1	7.83	7.69	0.00
uk	0	5	1	8	0	7	30.01	29.85	0.00
power	2	15	1	3	0	1	1.83	1.65	0.00
add32	0	32	1	2	1	0	1.12	0.64	0.00
hep-th	5	36	1	3	1	0	4.79	3.55	0.00
whitaker3	0	9	0	2	1	0	47.18	46.99	0.00
crack	0	10	1	2	1	0	34.91	34.65	0.00
PGP	24	62	325	5	1	2	44.54	25.98	0.00
cs4	0	6	1	6	0	5	411.37	409.26	0.00

Table 4.5: Results for Group-1 instances with $s = 2$ and $\alpha = 0.5$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Regular	Total time (s)	s-club time (s)	Gap (%)
karate	8	3	41	17	11	3	0.16	0.01	0.00
dolphins	3	10	200	16	13	2	0.52	0.25	0.00
lesmis	10	8	90	18	16	0	0.52	0.04	0.00
polbooks	16	10	268	23	17	2	0.41	0.13	0.00
adjnoun	12	10	331	16	12	1	1.07	0.67	0.00
football	1	15	973,384	96	2	90	92.14	26.56	0.00
jazz	44	26	1,497,964	5,307	56	5,249	TL	1271.72	12.62
celegansn	23	21	2,214	13	7	2	6.18	5.61	0.00
celegansm	32	10	181	7	2	1	0.48	0.09	0.00
email	12	38	1,020	6	1	0	39.51	37.44	0.00
polblogs	125	42	452,127	138	1	116	TL	1976.77	2.19
netscience	3	21	206	3	1	0	1.20	0.04	0.00
add20	52	34	7,088	9	2	0	12.59	0.95	0.00
data	1	17	20	3	0	1	13.25	12.86	0.00
uk	0	5	1	8	0	7	29.07	28.87	0.00
power	2	15	1	3	0	1	2.01	1.59	0.00
add32	4	29	58	4	1	0	3.19	1.09	0.00
hep-th	18	29	412	4	2	0	10.95	6.34	0.00
whitaker3	0	9	1	2	1	0	47.00	46.72	0.00
crack	1	9	1	3	1	0	72.81	72.51	0.00
PGP	45	47	5,858	3	1	0	41.63	8.20	0.00
cs4	0	6	1	7	0	6	503.88	501.18	0.00

decreases, interdiction is cheaper and there are many more feasible solutions of high quality distributed across the BC tree, thereby resulting in far fewer BC nodes being pruned.

Tables 4.6, 4.7, and 4.8 report our results for 3-club interdiction with $\alpha = 2, 1$, and 0.5, respectively. The number of instances that are solved to optimality within the time limit are 14, 13, and 12 for $\alpha = 2, 1$, and 0.5, respectively. (By contrast, 20 out of the 22 graphs for all three values of α were solved to optimality for 2-club interdiction.) In general, we observe that the 3-club interdiction problem is significantly more difficult to solve than its 2-club counterpart. When solving the 3-club interdiction problem using Method 3, we invoke separation more frequently and each callback to the separation problem takes more time to finish.

During separation, the maximum 3-club problem takes more time to solve than the maximum 2-club problem on our testbed (see Table 4.1 for instances where the difference is significant). But more importantly, on each maximum 3-club we find, the algorithm now

Table 4.6: Results for Group-1 instances with $s = 3$ and $\alpha = 2$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)	Gap (%)
karate	5	6	1	8	4	0.32	0.00	0.02	0.00
dolphins	3	19	11,011	95	93	4.49	1.86	0.55	0.00
lesmis	8	11	72	28	20	0.79	0.02	0.18	0.00
polbooks	10	25	50,926	192	187	20.54	7.18	1.94	0.00
adjnoun	10	25	497,453	774	767	429.93	208.39	13.00	0.00
football	2	50	11,670	120	116	232.65	220.00	5.16	0.00
jazz	5	145	45,760	3,633	3,629	TL	549.44	2646.06	40.17
celegansn	28	68	24,388	1,822	1,814	TL	2375.78	1164.87	37.85
celegansm	22	29	5,982	30	28	52.81	2.10	8.98	0.00
email	140	94	1	26	24	TL	3752.68	79.68	81.84
polblogs	340	228	1	20	19	TL	733.57	3180.50	81.93
netscience	6	27	155	8	6	2.78	0.12	0.04	0.00
add20	61	125	3,816	338	330	TL	77.85	787.69	53.32
data	0	32	1	8	6	45.97	43.34	0.04	0.00
uk	0	8	1	5	3	21.87	21.56	0.01	0.00
power	1	27	1	7	5	5.57	4.50	0.02	0.00
add32	5	75	729,074	65	63	TL	15.86	1.03	1.00
hep-th	0	120	1	83	82	TL	3573.15	4.91	44.66
whitaker3	0	15	1	6	4	177.24	175.41	0.01	0.00
crack	0	17	1	6	5	204.25	201.43	0.02	0.00
PGP	4	266	1	104	103	TL	2955.97	40.22	56.79
cs4	0	12	1	6	4	653.00	648.15	0.02	0.00

solves the latency-3 CDS problem as opposed to the heuristic used for $s = 2$. We find that the instances that were not solved to optimality also typically have significantly larger running times for finding a latency-3 CDS, compared to those instances that we do solve to optimality.

The number of calls to the separation routine, the number of cuts added, and the number of BC nodes have increased on average when compared to what is observed for $s = 2$. One possible explanation for this behavior is that the initial strength of the relaxation problem based on s-clubs in $\mathcal{S}^0$ is not as strong when $s = 3$ compared to when $s = 2$. In other words, the edge star based constraints (4.2) when $s = 3$ are possibly not as strong as star based constraints (4.1) when $s = 2$. The relative weakness of the initial relaxation based on edge star constraints may be due to large 3-clubs in the graph that do not resemble edge stars, while it is more common for large 2-clubs to resemble stars.

As solving the separation problem for both values of s requires a significant proportion of the overall solution time, we have evaluated the effect of using heuristics to solve the separation problem. Our results show that this approach might improve the performance of

Table 4.7: Results for Group-1 instances with $s = 3$ and $\alpha = 1$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)	Gap (%)
karate	6	5	17	7	3	0.19	0.00	0.02	0.00
dolphins	9	11	11,187	188	184	8.01	3.93	0.82	0.00
lesmis	8	11	84	9	6	0.41	0.00	0.07	0.00
polbooks	20	12	16,007	74	69	12.21	1.80	0.58	0.00
adjnoun	17	15	237,311	269	263	168.81	59.20	2.26	0.00
football	5	45	82,949	2,496	2,491	TL	3241.66	72.25	46.32
jazz	72	28	63,683	6,165	6,159	TL	1003.51	1751.19	33.32
celegansn	40	43	21,710	2,052	2,044	TL	3009.87	513.28	32.90
celegansm	29	19	21,846	34	30	78.16	1.31	8.82	0.00
email	167	70	1	46	45	TL	3639.63	93.73	74.15
polblogs	350	188	1	24	22	TL	850.14	3217.57	76.88
netscience	6	27	1,887	8	6	6.67	0.12	0.04	0.00
add20	102	47	194,989	4,242	4,235	TL	612.74	1549.38	35.94
data	1	31	32,189	28	25	390.48	182.72	0.19	0.00
uk	0	8	1	7	6	33.21	32.74	0.02	0.00
power	3	25	2,774	18	16	26.67	13.04	0.07	0.00
add32	16	55	935,727	92	89	TL	19.22	1.13	7.99
hep-th	6	114	1	93	91	TL	3457.43	4.33	53.44
whitaker3	0	15	1	7	6	212.12	209.15	0.02	0.00
crack	0	17	1	7	6	269.91	265.80	0.02	0.00
PGP	7	252	1	86	84	TL	2425.85	29.35	63.97
cs4	1	10	1	7	6	749.78	742.57	0.02	0.00

Method 3 depending on the test bed; see Section 4.3.3 for more details.

We close this section by noting that similar to the $s = 2$ case, the optimal value of θ shows that our model decreases the maximum 3-club size significantly except for those cases where $\bar{\omega}_3(G)$ is small. As before, the interdiction problem becomes more difficult to solve when the value of α is decreased.

4.2.3 Results for Group-2 instances

We evaluate the performance of Method 3 on Group-2 instances in this section. Tables 4.9, 4.10, and 4.11 show the results for $s = 2$. As mentioned before, graph `Douban` is not included in these experiments because the maximum 2-club for this instance is not found within the time limit. The results on the remaining 17 instances show that all of them are solved to optimality within the one hour time limit except instance `Anybeat` with $\alpha = 0.5$, which has a 1% relative optimality gap at termination. We also find that the value of $\bar{\omega}_2(G)$ remains the same for three instances `G05,G08,G09` with $\alpha = 2$, but in all other cases the 2-club number

Table 4.8: Results for Group-1 instances with $s = 3$ and $\alpha = 0.5$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)	Gap (%)
karate	7	4	78	2	0	0.24	0.00	0.00	0.00
dolphins	18	5	10,654	52	39	3.79	0.45	0.16	0.00
lesmis	13	7	85	4	0	0.59	0.00	0.00	0.00
polbooks	27	8	42,088	44	34	24.74	0.83	0.20	0.00
adjnoun	22	10	148,295	86	75	117.51	9.64	0.45	0.00
football	40	19	249,850	6,843	6,839	TL	2548.35	81.29	36.66
jazz	90	14	546,993	1,606	1,596	TL	230.80	16.60	19.85
celegansn	58	27	174,674	2,364	2,356	TL	2374.93	70.85	27.40
celegansm	35	14	279,199	60	55	717.01	4.37	0.81	0.00
email	251	50	1	56	54	TL	3671.76	143.89	71.90
polblogs	454	62	1	45	43	TL	1357.63	2118.54	68.46
netscience	12	21	7,867	8	6	30.69	0.12	0.04	0.00
add20	116	33	625,508	615	610	TL	45.23	108.92	24.90
data	1	31	471,772	37	33	TL	258.05	0.24	3.55
uk	0	8	1	9	8	51.10	50.15	0.03	0.00
power	7	22	20,025	23	20	114.11	16.14	0.08	0.00
add32	38	39	442,630	112	108	TL	18.63	0.98	14.14
hep-th	11	110	1	99	98	TL	3107.94	1.95	59.10
whitaker3	0	15	1	9	8	294.46	284.14	0.02	0.00
crack	0	17	1	10	9	405.37	394.57	0.03	0.00
PGP	10,680	0	1	89	88	TL	1783.37	41.73	98.68
cs4	1	10	1	9	8	1005.55	994.03	0.03	0.00

significantly decreases after interdiction. For example, the maximum 2-club size of `AS02` is $2,344$, while after interdiction it decreases to 114, 80, and 58, respectively, for α equal to $2, 1$, and 0.5.

Although during initialization of the relaxation we add star based constraints only for the top 20% of vertices by degree, as described in Section 4.1.1, the number of violated constraints that are added on-the-fly is never more than 4 (`G08` when $\alpha = 0.5$). As it can be seen under the columns Star and Leaf in the tables, in the vast majority of instances the largest 2-club found in the interdicted graph is frequently a star and our heuristic never added a constraint using just the leaves detected in H.

Similar to Group-1 instances, the interdiction problem becomes more difficult to solve for smaller values of α, and the number of BC nodes explored and the running time increase noticeably. As an example, the number of explored nodes for instance `Anybeat` increases from 136 when $\alpha = 2$ to $321,054$ when $\alpha = 0.5$. Moreover, the average running time for the 16 instances that are solved to optimality, increases from 116 seconds to 356 seconds as α

Table 4.9: Results for Group-2 instances with $s = 2$ and $\alpha = 2$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Regular	Total time (s)	s-club time (s)	Gap (%)
G04	2	67	1	3	1	0	49.19	47.22	0.00
G05	0	89	1	2	1	0	15.79	13.77	0.00
G06	1	74	1	3	1	0	16.75	15.68	0.00
G08	0	98	25	2	1	0	21.81	19.38	0.00
G09	0	103	26	2	1	0	23.65	19.99	0.00
B-Alpha	23	99	134	3	1	0	21.87	17.16	0.00
B-OTC	35	103	89	3	1	0	33.70	27.3	0.00
AS01	42	73	159	3	1	0	68.76	45.63	0.00
AS02	40	114	209	3	1	0	69.64	47.07	0.00
Ning	30	130	551	3	1	0	123.45	113.60	0.00
Hamsterster	19	89	65	3	0	1	23.07	20.65	0.00
Escorts	21	120	167	3	1	0	104.85	100.16	0.00
Anybeat	54	136	2,602	4	2	0	399.65	283.61	0.00
Advogato	40	131	300	3	1	0	1219.16	1204.89	0.00
Gplus	100	40	279	4	1	0	24.31	16.39	0.00
Facebook1	116	1	1	3	1	0	55.15	51.04	0.00
Facebook2	10	1	1	4	2	0	0.30	0.23	0.00

decreases from 2 to 0.5.

For $s = 3$, as mentioned before, solving the maximum s-club problem to optimality is too time-consuming for instances in Group-2 (See Table 4.2). Therefore, we use an inexact approach to solve the separation problem to find a sufficiently violated constraint (i.e., corresponding s-club) instead of finding a maximum s-club. Given an integral feasible solution $(\hat{\theta}, \hat{x})$ to the initial relaxation, instead of finding a maximum s-club in the graph interdicted according to $\hat{x}$, we look for an s-club with cardinality at least $\hat{\theta} + \epsilon$ where ϵ is the minimum violation we seek in the constraint.

In this inexact separation approach, first we rely on the greedy heuristic built into the ICUT solver to detect a sufficiently large s-club. If this heuristic s-club size is at least $\hat{\theta} + \epsilon$, the separation call is terminated early and the corresponding violated constraint is added to the initial problem. If the heuristic s-club is not sufficiently large, the exact Gurobi BC algorithm in the ICUT solver is run with a termination condition based on a target objective value. In this setting, the solver stops once it finds an s-club of size at least $\hat{\theta} + \epsilon$. If neither of the above two conditions results in early termination of ICUT, we let it continue to solve the

Table 4.10: Results for Group-2 instances with $s = 2$ and $\alpha = 1$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Regular	Total time (s)	s-club time (s)	Gap (%)
G04	2	67	134	3	1	0	50.07	46.66	0.00
G05	8	79	250	4	2	0	40.00	35.24	0.00
G06	5	69	60	3	1	0	25.07	19.60	0.00
G08	2	94	640	6	4	0	104.19	90.72	0.00
G09	20	78	1,431	3	1	0	49.13	30.18	0.00
B-Alpha	50	60	294	3	1	0	17.18	9.84	0.00
B-OTC	54	76	3,633	4	1	0	67.38	38.39	0.00
AS01	54	60	922	3	2	0	73.27	44.99	0.00
AS02	62	80	1,642	3	2	0	104.54	47.17	0.00
Ning	45	106	3,140	5	1	0	360.93	335.74	0.00
Hamsterster	25	82	614	4	0	1	43.33	40.20	0.00
Escorts	37	91	486	3	1	0	106.75	100.68	0.00
Anybeat	83	99	19,374	4	2	0	507.91	224.23	0.00
Advogato	62	106	5,868	6	2	0	2121.36	2027.90	0.00
Gplus	124	5	1	5	1	0	23.25	17.95	0.00
Facebook1	116	1	1	3	1	0	48.98	45.28	0.00
Facebook2	10	1	1	4	2	0	0.28	0.22	0.00

separation problem to optimality. In this case, it will terminate either returning a maximum s-club with violation, i.e., of size greater than $\hat{\theta}$ and smaller than $\hat{\theta} + \epsilon$; or certifying that no violated constraint exists. Note that by design, on our test bed ICUT subproblems do not reach their termination by time limit. After experimentation with $\epsilon = 1.5, 2.5$, and 5 in this inexact separation approach (see Section 4.3.3), we chose to employ a minimum constraint violation target of 1.5 for early termination of a separation call.

Moreover, instead of solving the minimum latency-s CDS problem to optimality, we use the following method that is analogous to Algorithm 3 to heuristically find a hereditary subset of the violated 3-club: if a 3-club S contains an edge $\{u, v\}$ such that $\deg_{G[S]}(u) + \deg_{G[S]}(v) - |c_{uv}| = |S|$ where $|c_{uv}|$ is the number of common neighbors of vertices u and v, then $\{u, v\} \subseteq S$ is a minimum latency-3 CDS of $G[S]$ and S is a H-hereditary 3-club for $H = S \setminus \{u, v\}$. Otherwise, we set $H = \{u \in S \mid \deg_{G[S]}(u) = 1\}$. Table 4.12 shows the results of these experiments for $\alpha = 2$. As it can be seen, only 3 instances `Gplus`, `Facebook1`, and `Facebook2` are solved to optimality within the time limit. We should remind the reader here that all separation calls terminated conclusively even though the *cumulative* separation time

Table 4.11: Results for Group-2 instances with $s = 2$ and $\alpha = 0.5$ using Method 3.

Graph G	$x(V)$	θ	#BC nodes	#CB	Star	Regular	Total time (s)	s-club time (s)	Gap (%)
G04	26	47	479	4	1	0	158.12	150.56	0.00
G05	63	31	1,909	4	1	0	127.67	115.26	0.00
G06	69	31	2,577	3	1	0	80.47	53.41	0.00
G08	75	26	476	3	1	0	41.43	37.97	0.00
G09	72	30	3,052	3	1	0	54.92	46.99	0.00
B-Alpha	81	41	36,709	6	2	0	160.82	20.72	0.00
B-OTC	82	54	37,748	4	1	0	239.13	21.59	0.00
AS01	73	44	4,687	3	1	0	94.28	47.02	0.00
AS02	96	58	31,063	8	2	2	674.79	171.61	0.00
Ning	93	75	99,359	4	1	0	908.55	171.84	0.00
Hamsterster	60	51	53,082	8	1	3	213.56	75.75	0.00
Escorts	79	65	18,715	3	1	0	212.38	80.37	0.00
Anybeat	118	71	321,054	6	3	0	TL	224.25	1.00
Advogato	102	80	154,273	9	3	0	2622.67	1971.76	0.00
Gplus	129	2	1	4	2	0	19.54	16.87	0.00
Facebook1	116	1	1	5	2	0	86.24	80.73	0.00
Facebook2	10	1	1	4	2	0	0.29	0.22	0.00

exceeds one hour in these instances. Since our previous experiments show that the interdiction problem becomes more difficult to solve on this test bed as the value of α decreases, we have not conducted experiments for $\alpha = 1$ and $\alpha = 0.5$. The results for the Group-2 instances reinforce the conclusions from our experiments with Group-1, that for $s = 3$ the interdiction problem becomes much more challenging to solve.

4.3 Additional experimental results

4.3.1 Comparison of root node performance of Method 1 and Method 3

We have compared the performance of Method 1 and Method 3 in the root node by setting a termination condition on the number of explored nodes. All other solver parameters including primal heuristics and general purpose cutting planes are at their default settings. With this condition, the solver terminates when the total number of branch-and-cut nodes explored exceeds the value specified in the Gurobi `NodeLimit` parameter (which is 1 in our case).

Tables 4.13 and 4.14 show the results of these experiments. Comparing the quality of the objective values and gaps obtained by each method in the root node shows that except for

Table 4.12: Results for Group-2 instances with $s = 3$ using inexact separation.

Graph G	$x(V)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)	Gap (%)
G04	10,876	0	1	5	4	TL	4,967.81	0.00	99.62
G05	8,846	0	1	9	8	TL	4,671.76	0.00	99.48
G06	8,717	0	1	87	86	TL	6,325.01	0.03	99.44
G08	5,672	9	1	135	133	TL	3,493.11	0.17	98.97
G09	8,114	0	1	57	56	TL	4,449.13	0.06	99.33
B-Alpha	3,783	0	1	117	116	TL	78.47	0.31	98.02
B-OTC	5,295	8	1	90	88	TL	148.10	0.24	98.32
AS01	62	82	1,050	125	122	TL	195.56	0.06	15.74
AS02	10,900	0	1	74	73	TL	178.72	0.52	99.11
Ning	10,298	0	1	47	46	TL	442.33	0.98	99.00
Hamsterster	1,674	7	18,099	3,386	3,384	TL	2,777.36	9.15	95.26
Escorts	10,106	0	1	5	3	TL	4,002.30	0.00	99.20
Anybeat	12,645	0	1	25	24	TL	190.87	0.73	99.10
Advogato	6,551	0	1	46	45	TL	559.49	1.16	98.33
Gplus	100	41	380	8	5	2405.92	53.99	0.07	0.00
Facebook1	116	1	1	5	3	318.07	170.60	0.07	0.00
Facebook2	10	1	1	6	4	2.50	0.63	0.00	0.00
Douban	154,908	0	1	4	3	TL	4,261.36	0.03	99.92

graph `football` for $s = 2$, Method 3 gives the same or a smaller gap and a smaller objective value than Method 1 at the root node. These results (along with the results presented in section 4.3.2) suggest that the improvements observed in Method 3 are predominantly because the heredity-based formulation is better than the standard formulation.

4.3.2 Impact of Gurobi heuristics on Method 3

As our DBC algorithm only separates integral solutions, it stands to reason that its performance will depend on the effectiveness of the primal heuristics built into the Gurobi solver that produce integral solutions to the initial relaxation. In order to examine the dependency of our algorithm performance on Gurobi primal heuristics, we have performed experiments that disable these heuristics.

Table 4.15 and 4.16 report the results for $s = 2$ and $s = 3$, respectively. When $s = 2$, we find that 16 out of 20 graphs are solved faster when turning off the primal heuristics (47% decrease on average), while the running times increase for other instances `adjnoun` (24%), `football` (122%), `celegansn` (55%) and `PGP` (16%). It can also be seen that in general, the number of explored nodes increases, and the number of callbacks and cuts decreases when

Table 4.13: Root node comparison of Method 1 and Method 3 on Group-1 instances for $s = 2$ and $\alpha = 0.5$.

Graph G	Method	#CB	#Cuts	Total time (s)	s-club time (s)	Obj Val	Gap (%)
karate	3	17	14	0.14	0.02	7.00	8.52
	1	84	82	0.26	0.12	16.50	92.42
dolphins	3	12	11	0.39	0.20	11.50	20.66
	1	45	44	1.78	1.66	12.00	83.33
lesmis	3	18	16	0.32	0.02	13.00	13.89
	1	47	43	0.15	0.07	21.50	93.02
polbooks	3	19	15	0.41	0.11	18.00	18.80
	1	51	49	1.20	1.11	24.50	91.84
adjnoun	3	12	9	0.90	0.59	16.00	20.11
	1	81	77	2.52	2.27	28.50	93.97
football	3	42	38	11.36	11.14	16.00	20.54
	1	34	29	10.12	10.03	15.50	77.88
celegansn	3	9	5	4.59	4.36	32.50	21.77
	1	40	39	4.82	4.07	135.00	98.12
celegansm	3	7	3	0.44	0.08	26.00	9.79
	1	36	35	0.66	0.51	238.00	98.74
email	3	5	1	24.35	23.65	44.50	9.76
	1	29	27	153.49	152.52	66.50	86.29
netscience	3	3	1	1.16	0.04	22.50	5.63
	1	73	71	2.14	1.27	29.50	49.37
add20	3	7	1	2.60	0.77	62.50	20.01
	1	98	97	93.01	18.97	124.00	86.88
data	3	3	1	12.59	12.19	17.50	4.62
	1	3	1	13.76	13.45	17.50	9.36
uk	3	8	7	27.04	26.85	5.00	0.00
	1	7	6	23.97	23.77	5.00	0.00
power	3	3	1	2.05	1.61	16.00	0.00
	1	36	34	27.39	26.16	16.00	5.52
add32	3	4	1	3.21	1.15	31.00	3.95
	1	54	53	23.96	13.78	32.00	33.55
hep-th	3	4	2	10.55	6.40	38.00	7.63
	1	163	162	465.74	267.26	51.00	48.35
whitaker3	3	2	1	44.94	44.68	9.00	0.00
	1	2	1	48.05	47.72	9.00	0.00
crack	3	3	1	69.49	69.19	9.50	0.00
	1	3	1	88.99	88.58	9.50	0.00
PGP	3	3	1	28.05	8.29	69.50	9.68
	1	138	137	933.31	636.85	206.00	84.89
cs4	3	7	6	716.82	713.64	6.00	0.00
	1	8	7	666.96	663.10	6.00	0.00

Table 4.14: Root node comparison of Method 1 and Method 3 on Group-1 instances for $s = 3$ and $\alpha = 0.5$.

Graph G	Method	#CB	#Cuts	Total time(s)	s-club time(s)	LCDS time(s)	Obj Val	Gap (%)
karate	3	2	0	0.24	0.00	0.00	7.50	5.89
	1	62	59	0.16	0.05		15.00	82.14
dolphins	3	33	30	1.07	0.41	0.24	17.50	34.99
	1	66	66	0.84	0.63		20.50	70.95
lesmis	3	4	0	0.6	0.00	0.00	13.50	7.81
	1	81	80	0.32	0.10		25.50	79.88
polbooks	3	35	31	2.88	0.79	0.32	22.50	22.63
	1	42	41	0.50	0.22		29.00	78.45
adjnoun	3	53	49	8.7	6.80	0.59	27.50	39.84
	1	51	49	4.01	3.31		31.00	76.20
celegansm	3	55	50	16.77	4.18	0.87	31.50	13.63
	1	32	31	8.80	4.23		184.50	94.69
netscience	3	8	6	13.22	0.11	0.05	27.00	8.49
	1	113	111	7.56	1.86		40.00	56.01
uk	3	9	8	46.01	44.95	0.04	8.00	0.00
	1	13	11	51.20	50.12		8.00	0.00
power	3	23	20	27.96	14.72	0.11	25.50	5.22
	1	211	210	193.99	164.46		51.50	58.11
whitaker3	3	9	8	274.45	264.57	0.04	15.00	0.00
	1	14	12	348.51	339.10		15.00	0.00
crack	3	10	9	383.09	372.56	0.04	17.00	0.00
	1	18	16	606.28	595.34		17.00	0.00
cs4	3	9	8	961.8	948.99	0.04	10.50	0.00
	1	14	14	1172.67	1158.99		10.50	0.00

Gurobi heuristics are turned off. When $s = 3$ the decrease in the running times is 39% on average for 9 out of 12 instances and for graphs `adjnoun`, `power` and `crack`, the running times increase 6%, 24%, and 4%, respectively. The number of explored nodes increases for 6 instances while the number of callbacks and cuts increase only for the graph `power`.

Based on these results, it is difficult to conclude that turning Gurobi heuristics on or off leads to a consistent, predictable impact on the overall performance. This may be attributed to the conflicting forces at play. Turning off Gurobi heuristics can result in fewer (or no) integral solutions encountered at the root node that invoke separation calls, with less time spent finding s-clubs and re-solving node relaxations as a result. In some (easier) instances,

Table 4.15: Impact of Gurobi heuristics on Method 3 when solving Group-1 instances for $s = 2$ and $\alpha = 0.5$.

Graph G	Heuristics	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)
karate	Off	8	3	80	17	15	0.07	0.01
	On	8	3	41	17	14	0.16	0.01
dolphins	Off	3	10	211	14	12	0.23	0.17
	On	3	10	200	16	15	0.52	0.25
lesmis	Off	10	8	335	18	16	0.06	0.01
	On	10	8	90	18	16	0.52	0.04
polbooks	Off	20	8	476	26	22	0.33	0.26
	On	16	10	268	23	19	0.41	0.13
adjnoun	Off	12	10	848	20	16	1.33	1.20
	On	12	10	331	16	13	1.07	0.67
football	Off	1	15	3,686,723	137	133	204.61	47.15
	On	1	15	973,384	96	92	92.14	26.56
celegansn	Off	23	21	3,290	18	14	9.57	9.18
	On	23	21	2,214	13	9	6.18	5.61
celegansm	Off	32	10	335	6	1	0.31	0.05
	On	32	10	181	7	3	0.48	0.09
email	Off	12	38	1,445	5	2	38.04	36.88
	On	12	38	1,020	6	1	39.51	37.44
netscience	Off	3	21	88	1	0	0.31	0.03
	On	3	21	206	3	1	1.20	0.04
add20	Off	52	34	4,701	8	1	8.98	0.83
	On	52	34	7,088	9	2	12.59	0.95
data	Off	1	17	28	1	0	5.10	4.84
	On	1	17	20	3	1	13.25	12.86
uk	Off	0	5	1	3	3	12.78	12.65
	On	0	5	1	8	7	29.07	28.87
power	Off	2	15	1	1	1	1.21	0.76
	On	2	15	1	3	1	2.01	1.59
add32	Off	4	29	56	1	0	0.90	0.26
	On	4	29	58	4	1	3.19	1.09
hep-th	Off	18	29	826	3	1	7.78	5.92
	On	18	29	412	4	2	10.95	6.34
whitaker3	Off	0	9	1	1	0	25.90	25.64
	On	0	9	1	2	1	47.00	46.72
crack	Off	1	9	1	1	0	32.36	32.08
	On	1	9	1	3	1	72.81	72.51
PGP	Off	45	47	4,613	4	0	48.14	17.10
	On	45	47	5,858	3	1	41.63	8.20
cs4	Off	0	6	1	4	3	363.48	361.44
	On	0	6	1	7	6	503.88	501.18

Table 4.16: Impact of Gurobi heuristics on Method 3 when solving Group-1 instances for $s = 3$ and $\alpha = 0.5$.

Graph G	Heuristics	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)
karate	Off	11	2	134	1	0	0.14	0.00	0.00
	On	7	4	78	2	0	0.24	0.00	0.00
dolphins	Off	20	4	7,344	21	19	1.86	0.27	0.08
	On	18	5	10,654	52	39	3.79	0.45	0.16
lesmis	Off	13	7	148	1	0	0.37	0.00	0.00
	On	13	7	85	4	0	0.59	0.00	0.00
polbooks	Off	27	8	47,587	21	18	16.84	0.10	0.07
	On	27	8	42,088	44	34	24.74	0.83	0.20
adjnoun	Off	26	8	181,072	29	27	124.49	3.15	0.11
	On	22	10	148,295	86	75	117.51	9.64	0.45
celegansm	Off	35	14	154,883	35	34	348.04	0.80	0.17
	On	35	14	279,199	60	55	717.01	4.37	0.81
netscience	Off	14	20	8,841	3	3	7.34	0.07	0.01
	On	12	21	7,867	8	6	30.69	0.12	0.04
uk	Off	2	7	1	6	6	44.45	42.54	0.03
	On	0	8	1	9	8	51.10	50.15	0.03
power	Off	7	22	25,879	31	29	141.99	30.96	0.16
	On	7	22	20,025	23	20	114.11	16.14	0.08
whitaker3	Off	0	15	1	6	6	238.44	221.16	0.03
	On	0	15	1	9	8	294.46	284.14	0.02
crack	Off	0	17	1	8	7	420.60	407.57	0.05
	On	0	17	1	10	9	405.37	394.57	0.03
cs4	Off	1	10	1	6	6	755.52	741.10	0.03
	On	1	10	1	9	8	1005.55	994.03	0.03

this can be beneficial as the wallclock time is reduced by simply letting the tree enumerate. However, on other instances, turning off Gurobi heuristics resulting in fewer integral solutions leading to fewer separation calls and fewer constraints generated at the root node, costs us in overall performance. A weaker relaxation at the root node and a larger tree size are a result of primal heuristic solutions not triggering constraint generation as often when turned off. However, the performance that is elicited by the choice seems to be very instance specific, and no doubt a function of the initial relaxation strength and integrality for the instance under consideration.

4.3.3 Impact of exact and inexact separation on Method 3

We performed experiments to evaluate the impact of inexact separation on the overall performance of Method 3. Given an integral feasible solution $(\hat{\theta}, \hat{x})$ to the initial relaxation, instead of finding a maximum s-club in the graph interdicted according to $\hat{x}$, we look for an s-club with cardinality at least $\hat{\theta} + \epsilon$ using the procedure described in Section 4.2.3, where ϵ is the minimum violation we seek in the constraint. We experimented with $\epsilon = 1.5$, 2.5, and 5.

Results of these experiments and their comparison with the default setting are shown in Tables 4.17–4.20. The last two columns of these tables, #ICUT-H and #ICUT-ϵ respectively indicate the number of lazy cuts detected using the first and second early termination attempts. Tables 4.17–4.19 report the results for $\epsilon = 5$, 2.5, and 1.5 when $s = 2$. As it can be seen, $\epsilon = 5$ is too large of a target for early termination and the separation problem is solved to optimality in most of the iterations of our test bed. As the value of ϵ decreases, more ϵ-violated cuts are found early. Tables 4.18 and 4.19 show that for both $\epsilon = 2.5$ and $\epsilon = 1.5$, the decrease in the running times is about 25% on average for 18 out of 20 instances. For the other 2 instances, the running times increase 4% and 6% on average for $\epsilon = 2.5$ and $\epsilon = 1.5$, respectively. Considering only those instances in Table 4.19 ($\epsilon = 1.5$) that take at least one minute to be solved, which are `football, crack` and `cs4`, we can observe that the running times of `football` and `crack` decrease 68% and 10% respectively, and the running time of `cs4` increases 10% when inexact separation is used. For all the other instances that take less than a minute to solve, the running times decrease at an average of 24%. Regarding the number of cuts in Table 4.19 ($\epsilon = 1.5$), for the graph `football`, 48 out of 55 total cuts are found by early termination. The total number of callbacks and cuts decreased from 96 and 92, respectively, to 60 and 55, which suggests that the cuts are sufficiently strong. For the graph `celegansn`, although all the violated constraints are found by heuristics, the total number of callbacks and cuts increased, which means that the cuts added using heuristics are weaker when compared to the cuts added by the exact solution in this instance. In other

Table 4.17: Inexact versus exact separation on Group-1 instances with $s = 2$, $\alpha = 0.5$, and $\epsilon = 5$.

Graph G	Method	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	#ICUT-H	#ICUT-ϵ
karate	inexact	8	3	41	17	14	0.15	0.02	0	0
	exact	8	3	41	17	14	0.16	0.01		
dolphins	inexact	3	10	200	16	15	0.38	0.18	3	1
	exact	3	10	200	16	15	0.52	0.25		
lesmis	inexact	10	8	90	18	16	0.31	0.01	3	0
	exact	10	8	90	18	16	0.52	0.04		
polbooks	inexact	16	10	268	23	19	0.46	0.11	6	0
	exact	16	10	268	23	19	0.41	0.13		
adjnoun	inexact	12	10	528	15	13	0.89	0.52	2	2
	exact	12	10	331	16	13	1.07	0.67		
football	inexact	1	15	356,392	56	51	28.40	10.89	19	0
	exact	1	15	973,384	96	92	92.14	26.56		
celegansn	inexact	23	21	1,528	24	20	2.80	2.37	20	0
	exact	23	21	2,214	13	9	6.18	5.61		
celegansm	inexact	30	11	181	7	3	0.45	0.06	1	0
	exact	32	10	181	7	3	0.48	0.09		
email	inexact	12	38	1,020	6	1	32.54	30.56	1	0
	exact	12	38	1,020	6	1	39.51	37.44		
netscience	inexact	3	21	206	3	1	1.34	0.03	1	0
	exact	3	21	206	3	1	1.20	0.04		
add20	inexact	52	34	7,088	9	2	12.51	0.87	0	0
	exact	52	34	7,088	9	2	12.59	0.95		
data	inexact	1	17	20	3	1	10.02	9.64	1	0
	exact	1	17	20	3	1	13.25	12.86		
uk	inexact	0	5	1	8	7	26.94	26.71	0	0
	exact	0	5	1	8	7	29.07	28.87		
power	inexact	2	15	1	3	1	1.86	1.40	0	0
	exact	2	15	1	3	1	2.01	1.59		
add32	inexact	4	29	58	4	1	3.02	0.96	0	0
	exact	4	29	58	4	1	3.19	1.09		
hep-th	inexact	18	29	412	4	2	10.66	5.95	1	0
	exact	18	29	412	4	2	10.95	6.34		
whitaker3	inexact	0	9	1	2	1	27.56	27.26	1	0
	exact	0	9	1	2	1	47.00	46.72		
crack	inexact	1	9	1	3	1	64.39	63.97	1	0
	exact	1	9	1	3	1	72.81	72.51		
PGP	inexact	45	47	5,858	3	1	40.49	7.42	1	0
	exact	45	47	5,858	3	1	41.63	8.20		
cs4	inexact	0	6	1	8	6	521.39	517.40	0	3
	exact	0	6	1	7	6	503.88	501.18		

Table 4.18: Inexact versus exact separation on Group-1 instances with $s = 2$, $\alpha = 0.5$, and $\epsilon = 2.5$.

Graph G	Method	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	#ICUT-H	#ICUT-ϵ
karate	inexact	8	3	41	17	14	0.10	0.01	8	0
	exact	8	3	41	17	14	0.16	0.01		
dolphins	inexact	3	10	218	16	14	0.33	0.16	9	3
	exact	3	10	200	16	15	0.52	0.25		
lesmis	inexact	10	8	90	18	16	0.27	0.01	10	0
	exact	10	8	90	18	16	0.52	0.04		
polbooks	inexact	16	10	268	23	19	0.39	0.10	16	0
	exact	16	10	268	23	19	0.41	0.13		
adjnoun	inexact	12	10	394	15	13	0.68	0.33	10	1
	exact	12	10	331	16	13	1.07	0.67		
football	inexact	1	15	510,258	65	60	38.98	12.73	22	8
	exact	1	15	973,384	96	92	92.14	26.56		
celegansn	inexact	23	21	1,528	24	20	2.61	2.21	20	0
	exact	23	21	2,214	13	9	6.18	5.61		
celegansm	inexact	30	11	181	7	3	0.42	0.06	1	0
	exact	32	10	181	7	3	0.48	0.09		
email	inexact	12	38	1,020	6	1	31.47	29.42	1	0
	exact	12	38	1,020	6	1	39.51	37.44		
netscience	inexact	3	21	206	3	1	1.21	0.03	1	0
	exact	3	21	206	3	1	1.20	0.04		
add20	inexact	52	34	7,088	9	2	12.42	0.85	1	0
	exact	52	34	7,088	9	2	12.59	0.95		
data	inexact	1	17	20	3	1	9.67	9.28	1	0
	exact	1	17	20	3	1	13.25	12.86		
uk	inexact	0	5	1	8	7	20.94	20.71	3	0
	exact	0	5	1	8	7	29.07	28.87		
power	inexact	2	15	1	3	1	1.83	1.35	1	0
	exact	2	15	1	3	1	2.01	1.59		
add32	inexact	4	29	58	4	1	3.11	0.95	1	0
	exact	4	29	58	4	1	3.19	1.09		
hep-th	inexact	18	29	412	4	2	10.64	5.93	1	0
	exact	18	29	412	4	2	10.95	6.34		
whitaker3	inexact	0	9	1	2	1	27.21	26.90	1	0
	exact	0	9	1	2	1	47.00	46.72		
crack	inexact	1	9	1	3	1	65.84	65.38	1	0
	exact	1	9	1	3	1	72.81	72.51		
PGP	inexact	45	47	5,858	3	1	40.66	7.44	1	0
	exact	45	47	5,858	3	1	41.63	8.20		
cs4	inexact	0	6	1	9	7	540.55	536.14	3	0
	exact	0	6	1	7	6	503.88	501.18		

Table 4.19: Inexact versus exact separation on Group-1 instances with $s = 2$, $\alpha = 0.5$, and $\epsilon = 1.5$.

Graph G	Method	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	#ICUT-H	#ICUT-ϵ
karate	inexact	8	3	41	17	14	0.09	0.01	9	0
	exact	8	3	41	17	14	0.16	0.01		
dolphins	inexact	3	10	215	16	14	0.39	0.19	12	0
	exact	3	10	200	16	15	0.52	0.25		
lesmis	inexact	10	8	90	18	16	0.28	0.01	11	0
	exact	10	8	90	18	16	0.52	0.04		
polbooks	inexact	16	10	268	23	19	0.32	0.03	18	0
	exact	16	10	268	23	19	0.41	0.13		
adjnoun	inexact	12	10	433	18	16	0.66	0.33	14	0
	exact	12	10	331	16	13	1.07	0.67		
football	inexact	1	15	503,717	60	55	29.72	5.86	23	25
	exact	1	15	973,384	96	92	92.14	26.56		
celegansn	inexact	23	21	1,528	24	20	2.60	2.19	20	0
	exact	23	21	2,214	13	9	6.18	5.61		
celegansm	inexact	30	11	181	7	3	0.41	0.06	1	0
	exact	32	10	181	7	3	0.48	0.09		
email	inexact	12	38	1,020	6	1	31.40	29.49	1	0
	exact	12	38	1,020	6	1	39.51	37.44		
netscience	inexact	3	21	206	3	1	1.21	0.03	1	0
	exact	3	21	206	3	1	1.20	0.04		
add20	inexact	52	34	7,088	9	2	12.40	0.88	1	0
	exact	52	34	7,088	9	2	12.59	0.95		
data	inexact	1	17	20	3	1	9.71	9.34	1	0
	exact	1	17	20	3	1	13.25	12.86		
uk	inexact	0	5	1	8	7	21.04	20.82	3	0
	exact	0	5	1	8	7	29.07	28.87		
power	inexact	2	15	1	3	1	1.80	1.35	1	0
	exact	2	15	1	3	1	2.01	1.59		
add32	inexact	4	29	58	4	1	3.10	0.94	1	0
	exact	4	29	58	4	1	3.19	1.09		
hep-th	inexact	18	29	412	4	2	10.86	6.01	1	0
	exact	18	29	412	4	2	10.95	6.34		
whitaker3	inexact	0	9	1	2	1	27.85	27.56	1	0
	exact	0	9	1	2	1	47.00	46.72		
crack	inexact	1	9	1	3	1	65.88	65.50	1	0
	exact	1	9	1	3	1	72.81	72.51		
PGP	inexact	45	47	5,858	3	1	40.18	7.43	1	0
	exact	45	47	5,858	3	1	41.63	8.20		
cs4	inexact	0	6	1	9	7	555.40	550.93	3	0
	exact	0	6	1	7	6	503.88	501.18		

Table 4.20: Inexact versus exact separation on Group-1 instances with $s = 3$, $\alpha = 0.5$, and $\epsilon = 1.5$.

Graph G	Method	$x(v)$	θ	#BC nodes	#CB	#Cuts	Total time (s)	s-club time (s)	LCDS time (s)	#ICUT-H	#ICUT-ϵ
karate	inexact	7	4	78	2	0	0.21	0.00	0.00	0	0
	exact	7	4	78	2	0	0.24	0.00	0.00		
dolphins	inexact	18	5	9,137	66	56	3.46	0.24	0.28	30	3
	exact	18	5	10,654	52	39	3.79	0.45	0.16		
lesmis	inexact	13	7	85	4	0	0.54	0.00	0.00	0	0
	exact	13	7	85	4	0	0.59	0.00	0.00		
polbooks	inexact	27	8	70,436	56	49	29.27	0.50	0.37	22	3
	exact	27	8	42,088	44	34	24.74	0.83	0.20		
adjnoun	inexact	24	9	189,962	119	107	119.30	5.07	0.66	35	33
	exact	22	10	148,295	86	75	117.51	9.64	0.45		
celegansm	inexact	35	14	98,595	39	37	353.80	1.62	1.06	9	18
	exact	35	14	279,199	60	55	717.01	4.37	0.81		
netscience	inexact	12	21	7,867	8	6	30.70	0.07	0.06	4	0
	exact	12	21	7,867	8	6	30.69	0.12	0.04		
uk	inexact	0	8	1	9	8	36.30	35.24	0.04	0	0
	exact	0	8	1	9	8	51.10	50.15	0.03		
power	inexact	7	22	17,530	24	22	105.00	16.35	0.12	8	3
	exact	7	22	20,025	23	20	114.11	16.14	0.08		
whitaker3	inexact	0	15	1	9	8	142.25	131.67	0.04	1	5
	exact	0	15	1	9	8	294.46	284.14	0.02		
crack	inexact	0	17	1	10	9	192.31	176.81	0.05	0	6
	exact	0	17	1	10	9	405.37	394.57	0.03		
cs4	inexact	1	10	1	9	8	581.47	568.57	0.04	0	6
	exact	1	10	1	9	8	1005.55	994.03	0.03		

instances, the number of callbacks and total number of cuts are the same for both exact and inexact separation or the difference is negligible.

The results of the experiments for the 3-club interdiction problem showed similar behavior, thus we only report the results for $\epsilon = 1.5$, the case where more violated constraints are found by inexact separation. In Table 4.20, out of 12 instances, the running times decrease 31% on average for six instances and increase about 7% on average for three instances.

Comparing the decrease in the running times for all the instances (25% when $s = 2$ and 31% when $s = 3$ for $\epsilon = 1.5$) with the decrease in the more challenging instances that take at least a minute to solve (39% when $s = 2$ and 41% when $s = 3$ for $\epsilon = 1.5$) shows that using inexact separation is more helpful for solving the more challenging instances.

CHAPTER V

FIRST PASSAGE TIME INTERDICTION

In network interdiction problems, we cannot always assume that evaders act deterministically and seek an optimal strategy to achieve their goal. Evaders might be unpredictable and choose their actions randomly due to incomplete information about the network and uncertainties or limited time for finding an optimal solution. In these situations, we can describe the behavior of the evaders with a Markov chain. In this chapter, we introduce a version of the network interdiction problem with markovian evaders where at each step, he/she chooses the next node randomly to build their path toward a target and the interdictor's goal is to increase the time taken the evader to reach their target. The motivation for this setting is its application in online social networks where malicious accounts spread harmful content in the network quickly and it is difficult for the social network manager or informed users to take an action. In the next section, we discuss the problem and our approach to solve it.

5.1 Problem statement

Following Berkhout and Heidergott (2019), consider a directed graph $G = (V, E)$ that represents the connections in the social network such that V is the set of users and there is a directed arc $(i, j) \in E$ if user j follows user i in the network. The graph also contains a loop for every vertex $i \in V$, i.e., $(i, i) \in E$. We model the exposure of any given harmful post to a user using a discrete time Markov chain (DTMC) and assume that the DTMC is irreducible. We define a Markov chain with transition probability matrix P on graph G such that $P_{ij} > 0$ for every $(i, j) \in E$ and $P_{ij} = 0$ if $(i, j) \notin E$. Let X_n denote the state of the chain at time n, which in the social network, shows the location of a message, e.g., $X_n = i$ means that at time n, user i receives the message. We use a random walk model to interpret matrix P: if the walk is at state i at time n, it reaches state j with probability P_{ij} at time $n + 1$, meaning that from user i the message jumps to user j with probability P_{ij}, thus the state of the message at time $n + 1$ will be j.

We define the expected first passage time in the social network as the first time user j observes a post that has been shared in the network by user i. Assuming a transition probability matrix is given, the interdictor has the information on the chance that the message moves from a user to their connections and is able to compute the expected first passage times between users by solving a system of linear equations.

Suppose the network contains a group of malicious users denoted by S and a group of vulnerable users denoted by T. The goal is to increase the expected first passage times from users in S to users in T. Figure 5.1 shows an example of this setting where $S = \{1\}$ and $T = \{4\}$. The parameter P_{ij} on each arc can be interpreted as the probability that, within a given and fixed time interval (e.g., a second), user i shares a post with user j and that user j reads the post. Also, for every vertex, there is a loop with a probability P_{ii} that shows the chance that the user decides not to share the post with their connections or that the user shares it but their connections do not read it. The interdictor must find a way to increase the expected time that it takes user 4 to learn about the harmful post, given that 1 is the initial user that knows about the message ($X_0 = 1$). This is possible by modifying the probabilities.

Since the goal is to increase the first passage times, in our setting, interdicting vertex $i \in V$ decreases the probability of traversing every arc (i, j) for every vertex $j \in N(i)$ where $N(i)$ is the set of neighbors of vertex i. Also, considering that the summation of the probabilities of the outgoing arcs and the loop for every vertex must be equal to one, interdicting vertex i increases the probability of traversing the loop from i to itself, and the increase is equal to

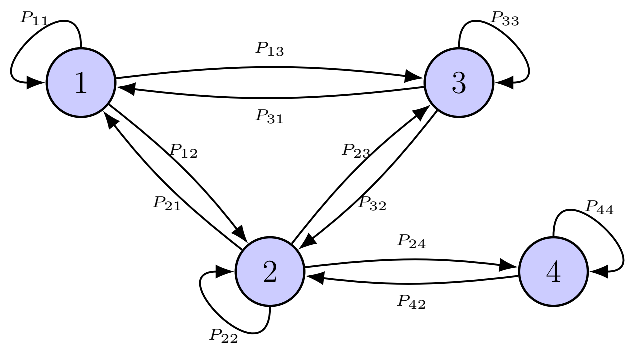

Figure 5.1: $S = \{1\}$ and $T = \{4\}$. The goal is to increase the expected first passage time from 1 to 4.

the total decrease in the probability of all the outgoing arcs of this vertex. Figure 5.2 shows the way probabilities are updated when a vertex is interdicted. By interdicting vertex i, the probability of traversing each arc (i, j) will decrease to $P_{ij}(1 - \Delta_{ij})$ where $0 \leq \Delta_{ij} \leq 1$ and $j \in N(i)$. Also, the probability of traversing the loop will increase to $P_{ii} + \sum_{j \in N(i)} P_{ij}\Delta_{ij}$.

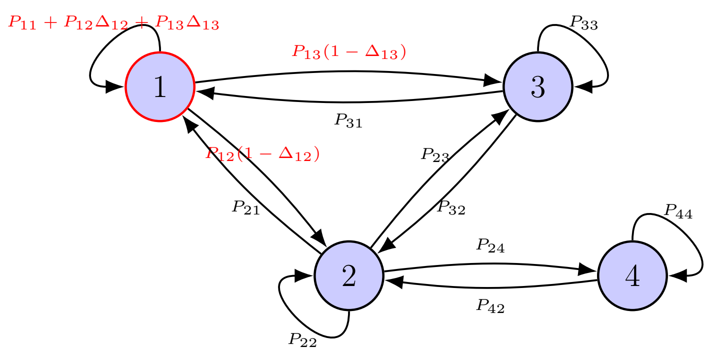

Figure 5.2: By interdicting vertex 1, value of P_{11} increases and value of P_{12} and P_{13} decrease.

Based on this setting, we study the optimization problem (5.1) where the goal is to find an optimal interdiction policy of size B to maximize the smallest expected first passage time from S to T:

$$\max \left\{ \min\{t_{ij}(x) : i \in S, j \in T\} : \sum_{i \in V} x_i \leq B \right\}. \tag{5.1}$$

In this problem, x_i is a binary variable that takes one if vertex i is interdicted and $t_{ij}(x)$ is the (expected) first passage time from i to j given that the nodes dictated by x are interdicted and can be computed using the following equation:

$$t_{ij}(x) = 1 + \sum_{\substack{k \in N(i) \\ k \neq j}} P_{ik}(1 - \Delta_{ik} x_i) t_{kj} + (P_{ii} + \sum_{k \in N(i)} P_{ik} \Delta_{ik} x_i) t_{ij} \quad \forall i, j \in V, i \neq j \tag{5.2}$$

Observe that Equation (5.2) is analogous to Equation (1.1), with the difference that it includes the effect of interdiction.

5.2 Reasonable values for Δ_{ij}

In this section, we determine the values that ensure that using Δ_{ij} to modify the probabilities will not result in decreasing the values of the first passage times.

Proposition 6. *If the interdiction penalties Δ_{ij} depend only on the departing state, i.e., if $\Delta_{ij} = \Delta_i$ for all $j \in N(i), j \neq i$, $i \in V$, then interdiction actions never decrease the first passage times.*

Proof. To this end, we consider a fixed state j and the first passage times to state j. Nominally (that is, pre-interdiction), the first passage times t_{ij}, $i \neq j$, satisfy

$$t_{ij} = \frac{1}{1 - P_{ii}} + \frac{\sum_{k \neq i,j} P_{ik} t_{ik}}{1 - P_{ii}}. \tag{5.3}$$

Now, assume $0 \leq \Delta_i < 1$ are given, $i \neq j$, and suppose that P_{ik} becomes $P_{ik}(1 - \Delta_i)$ for all $k \neq i$, and that P_{ii} becomes $P_{ii} + \sum_{k \neq i} \Delta_i P_{ik} = \Delta_i + (1 - \Delta_i) P_{ii}$. With the modified probabilities, it can be shown using the first passage times equations that the first passage times in the modified network, $\hat{t}_{ij}$, $i \neq j$ satisfy

$$\hat{t}_{ij} = \frac{1}{(1 - \Delta_i)(1 - P_{ii})} + \frac{\sum_{k \neq i,j} P_{ik} \hat{t}_{ik}}{1 - P_{ii}}. \tag{5.4}$$

Let N denote the number of vertices in the graph and B be the $(N-1)\times(N-1)$ matrix defined by

$$B_{qr} = \begin{cases} \frac{P_{qr}}{1-P_{qq}}, & \text{if } q,r \neq j,\ q \neq r \\ 0, & \text{if } q,r \neq j,\ q = r, \end{cases} \tag{5.5}$$

and let I be an $(N-1)\times(N-1)$ identity matrix. Furthermore, let u and w be the $(N-1)\times 1$ vectors defined by

$$u_q = \frac{1}{1-P_{qq}} \text{ and } w_q = \frac{1}{(1-\Delta_q)(1-P_{qq})} \qquad q \neq j. \tag{5.6}$$

Then, the vector forms of the first passage times equations are:

$$(I-B)V = u \text{ and } (I-B)\hat{V} = w. \tag{5.7}$$

where V is the vector with the t_{ij} and $\hat{V}$ is the vector with the $\hat{t}_{ij}$.

It is readily checked that the rows of matrix B sum to at most one. Moreover, at least one row sums to strictly less than one (otherwise, the probability of reaching state j will be zero). Thus, as all entries in B are non-negative, basic determinant properties imply that the determinant of B is strictly less than one (and strictly greater than zero if the DTMC is irreducible, which we assume). Thus, $(I-B)^{-1}$ exists and is given by

$$(I-B)^{-1} = \sum_{t=0}^{\infty} B^t. \tag{5.8}$$

Consequently,

$$V = \sum_{t=0}^{\infty} B^t u \text{ and } \hat{V} = \sum_{t=0}^{\infty} B^t w. \tag{5.9}$$

Since each element in B^t is non-negative for all $t \geq 0$ and since $w \geq u \geq 0$, it can be concluded that $\hat{V} \geq V$. This observation implies (as the Δ_{ij}s are arbitrary in $[0,1)$ and hence can be made zero), that any interdiction action never decreases the first passage times. Moreover, they imply that a valid lower bound for t_{ij}, for any interdiction decision, is the nominal value of the first passage times, that is, the first passage times in the network without any interdiction, and that an upper bound for t_{ij} is the value of the first passage times after all vertices are interdicted. $\square$

Now, using a counterexample, we show that if $\Delta_{ij} \neq \Delta_{ik}$ for some $j, k \in N(i)$, then by interdicting vertex i, first passage times might decrease. Consider Figure 5.3 and assume that for every $i \in V$, probabilities P_{ij} are equal to $\frac{1}{\deg(i)+1}$ if $j \in N[i]$ and 0 otherwise. With these probabilities, in the original graph, we have $t_{ij} = 4.5$ if $(i, j) \in E$ and $t_{ij} = 6$ if $(i, j) \notin E$. Suppose that vertex 1 is interdicted and $\Delta_{12} = 0.2$ and $\Delta_{13} = 0.9$. Computing the first passage times with these parameters shows that the values of the first passage times from vertices 1, 2, and 4 to vertex 3 respectively decrease to 3.72, 5.48, and 4.24.

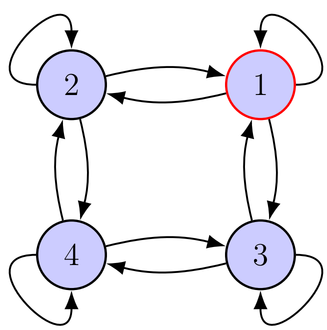

Figure 5.3: Counterexample to show different values of Δ_{ij} for $j \in N(i)$ can result in a decrease in the value of the first passage times.

Based on these observations, we can conclude that for every vertex $i \in V$, the value of Δ_{ij} must be equal for all $j \in N(i)$ to avoid interdiction reducing the values of the first passage times.

5.3 Complexity

In this section, we study the complexity of problem (5.1). We reduce from Vertex Cover to the decision problem of determining whether the first passage time from every vertex in S to every vertex in T can be raised to a certain threshold using at most B interdicted vertices. Given a vertex cover problem instance, i.e., an undirected graph $G = (V, E)$ and an integer B, we construct an instance of our interdiction problem on a directed graph $G' = (V, E')$. The graph G' extends graph G by adding a target vertex w, which is made adjacent to all other vertices, and by transforming each undirected arc into two directed arcs, as it is shown in Figure 5.4.

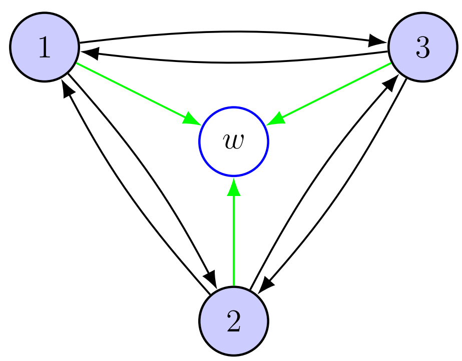

Figure 5.4: Undirected graph $G = \{V, E\}$ contains vertices $V = \{1, 2, 3\}$. Graph G' is obtained by transforming every undirected arc in G into two directed arcs and adding vertex w and arcs (i, w) for every $i \in V$.

Let $S = V$, $T = \{w\}$, and define the transition matrix P in graph G' be as follows where $d_i = \deg_G(i)$:

$$P_{ij} = \begin{cases} 0 & \text{if } i = j \\ p & \text{if } i \in V, j = w \\ (1-p)/d_i & \text{otherwise} \end{cases} \tag{5.10}$$

For a particular solution, we define the profit of a vertex as the value of the first passage time from that vertex to the target (vertex w). Theorem 5 shows the interdiction problem is NP-hard by reduction from the vertex cover problem.

Theorem 5. *The vertex cover instance admits a vertex cover of size B if and only if there exists a sufficiently large p, on graph G', with transition probability matrix P given by Equation* (5.10)*, with $\Delta_i = p$, for every $i \in V$, such that the value of Equation* (5.1) *is at least 2.*

Proof. By interdicting vertex i, probability of traversing arc (i,j) becomes $P_{ij}(1-\Delta_i)$ for every $i \in V$. Also, $N(i)$ in this section denotes $N_G(i)$ and does not include vertex w (subscript G is dropped for convenience.). First, assume there is a size-B vertex cover C of G. Then an interdiction solution where all the vertices in C are interdicted will have the following profit t_{iw} for each vertex $i \in V$:

$$t_{iw} = 1 + \sum_{\substack{k \in N(i) \\ k \neq w}} \frac{(1-p)}{d_i}(1-\Delta_i x_i)t_{kw} + \Big(\sum_{k \in N(i)} \frac{(1-p)}{d_i}\Delta_i x_i + p\Delta_i x_i\Big)t_{iw} \tag{5.11}$$

By replacing Δ_i with p, we have:

$$t_{iw} = \frac{1}{(1-px_i)} + \frac{(1-p)}{d_i}\sum_{\substack{k \in N(i) \\ k \neq w}} t_{kw} \tag{5.12}$$

Consider the following cases:

- Vertex $i \in C$: In this case, $x_i = 1$ and the value of the first passage time for any $0 \leq p < 1$ will be:

$$t_{iw} = \frac{1}{(1-p)} + \frac{(1-p)}{d_i}\sum_{\substack{k \in N(i) \\ k \neq w}} t_{kw} \tag{5.13}$$

Since C is a vertex cover and $i \in C$, to find the lower bound of t_{iw}, we can assume that

$k \notin C$ for every $k \in N(i)$, thus $x_k = 0$. By replacing t_{kw} with its equivalent value using Equation (5.12), where $x_k = 0$, we have:

$$t_{iw} = \frac{1}{(1-p)} + \frac{(1-p)}{d_i} \sum_{\substack{k \in N(i) \\ k \neq w}} \Big(1 + \frac{(1-p)}{d_k} \sum_{\substack{j \in N(k) \\ j \neq w}} t_{jw}\Big) \tag{5.14a}$$

$$= \frac{1}{(1-p)} + (1-p) + \frac{(1-p)^2}{d_i} \sum_{\substack{k \in N(i) \\ k \neq w}} \frac{1}{d_k} \sum_{\substack{j \in N(k) \\ j \neq w}} t_{jw} \tag{5.14b}$$

It can be seen that if $p \to 1$, then $t_{iw} \to \infty$ because of the first term in Equation (5.14b), so in this case, $t_{iw} \geq 2$.

- Vertex $i \notin C$: In this case, $x_i = 0$ and the value of the first passage time will be:

$$t_{iw} = 1 + \frac{(1-p)}{d_i} \sum_{\substack{k \in N(i) \\ k \neq w}} t_{kw} \tag{5.15}$$

Since C is a vertex cover, if $i \notin C$, then $k \in C$ for every $k \in N(i)$ and thus $x_k = 1$. By replacing t_{kw} with its equivalent value using Equation (5.13), we can see $t_{iw} \geq 2$:

$$t_{iw} = 1 + \frac{(1-p)}{d_i} \sum_{\substack{k \in N(i) \\ k \neq w}} \Big(\frac{1}{(1-p)} + \frac{(1-p)}{d_k} \sum_{\substack{j \in N(k) \\ j \neq w}} t_{jw}\Big) \tag{5.16a}$$

$$= 2 + \frac{(1-p)^2}{d_i} \sum_{\substack{k \in N(i) \\ k \neq w}} \frac{1}{d_k} \sum_{\substack{j \in N(k) \\ j \neq w}} t_{jw} \tag{5.16b}$$

We can conclude that if there is a vertex cover of size B and if p is sufficiently close to 1, then the maximum of the minimum first passage time from V to w after interdiction in G' has to be at least 2.

Now assume there is no size-B vertex cover. Therefore, for any interdiction set S of size B, there exists at least one arc (i, j) where none of the vertices i and j are interdicted. Recall that t_{ij} is the expected value of the first passage times, thus to compute the profit of vertex i we can write:

$$t_{iw} = P[t_{iw} = 1] + 2P[t_{iw} = 2] + E[t_{iw} | t_{iw} \geq 3] P[t_{iw} \geq 3]. \tag{5.17}$$

Note that $P[t_{iw} = 1] = p$ and

$$P[t_{iw} = 2] = \frac{(1-p)}{d_i} \sum_{\substack{j \in N(i) \\ j \notin C}} p + \frac{(1-p)}{d_i} \sum_{\substack{j \in N(i) \\ j \in C}} p(1-p) \tag{5.18a}$$

$$= \frac{p(1-p)d_i^-}{d_i} + \frac{p(1-p)^2 d_i^+}{d_i} = \frac{p(1-p)}{d_i}\big(d_i^- + (1-p)d_i^+\big). \tag{5.18b}$$

Moreover,

$$P[t_{iw} \geq 3] = 1 - P[t_{iw} = 1] - P[t_{iw} = 2] = (1-p)\left(1 - \frac{p(d_i^- + (1-p)d_i^+)}{d_i}\right)$$

where d_i^- and d_i^+ respectively show the cardinality of sets $\{j \in N(i) : j \notin C\}$ and $\{j \in N(i) : j \in C\}$, and $d_i^- \geq 1$ as C is not a vertex cover. It can be verified that if all the vertices are interdicted, then the value of the first passage time from i to w denoted by $\hat{t}_{iw}$ will be equal to $1/p(1-p)$. By exploiting the Markov property, this observation implies that:

$$E[t_{iw} | t_{iw} \geq 3] \leq \frac{1}{p(1-p)} \tag{5.19}$$

Therefore, we can write:

$$t_{iw} \leq p + \frac{1}{p} + \frac{2p(1-p) - 1}{d_i}\big(d_i^- + (1-p)d_i^+\big) \tag{5.20}$$

It can be seen that if $p \to 1$:

$$t_{iw} \leq 1 + 1 - \frac{d_i^-}{d_i} \tag{5.21}$$

Therefore, t_{iw} is strictly smaller than 2 since $d_i^- \geq 1$. We showed that if there is no size-B vertex cover, then, by making p sufficiently close to 1, there is at least one vertex with a profit of less than 2. This completes our proof and shows the problem is NP-hard. □

5.4 An MILP formulation

In this section, we present a mixed-integer Programming formulation of problem (5.1). We denote by t_{ij} the expected first passage time from i to j and by θ the smallest first passage time from S to T. Also, we use $x \in \{0,1\}^{|V|}$ to show the interdiction set such that $x_i = 1$ if vertex i is interdicted and zero otherwise:

$$z^0 = \max \ \theta \tag{5.22a}$$

$$\sum_{i \in V} x_i \leq B \tag{5.22b}$$

$$\theta \leq t_{ij} \qquad \forall\, i \in S, j \in T \tag{5.22c}$$

$$t_{ij} = 1 + \sum_{\substack{k \in N(i) \\ k \neq j}} P_{ik}(1 - \Delta_i x_i) t_{kj} + \Big(P_{ii} + \sum_{k \in N(i)} P_{ik} \Delta_i x_i\Big) t_{ij} \ \forall\, i, j \in V, i \neq j \tag{5.22d}$$

$$x_i \in \{0,1\} \qquad \forall\, i \in V \tag{5.22e}$$

$$t_{ij} \geq 0 \qquad \forall\, i, j \in V. \tag{5.22f}$$

Constraint (5.22c) requires θ to be smaller than any first passage time in the graph (between sets S and T) and constraint (5.22d) is built based on the equation (1.1) by considering the possibility of interdicting the vertices. This formulation is not linear because of the constraint (5.22d) and we introduce variable z_{ij} to linearize it:

$$z_{ij} = \Bigg(\sum_{k \in N(i)} P_{ik} \Delta_i t_{ij} - \sum_{\substack{k \in N(i) \\ k \neq j}} P_{ik} \Delta_i t_{kj} \Bigg) x_i \qquad \forall i, j \in V. \tag{5.23}$$

Using the new variables, the linear formulation will be as follows:

$$z^0 = \max \theta \tag{5.24a}$$

$$\sum_{i \in V} x_i \leq B \tag{5.24b}$$

$$\theta \leq t_{ij} \qquad \forall\, i \in S, j \in T \tag{5.24c}$$

$$t_{ij} = 1 - z_{ij} + \sum_{\substack{k \in N(i) \\ k \neq j}} P_{ik} t_{kj} + P_{ii} t_{ij} \qquad \forall\, i, j \in V, i \neq j \tag{5.24d}$$

$$z_{ij} \leq \sum_{\substack{k \in N(i) \\ k \neq j}} P_{ik} \Delta_i t_{kj} - \sum_{k \in N(i)} P_{ik} \Delta_i t_{ij} + M_{ij}(1 - x_i) \;\; \forall\, i, j \in V, i \neq j \tag{5.24e}$$

$$z_{ij} \geq \sum_{\substack{k \in N(i) \\ k \neq j}} P_{ik} \Delta_i t_{kj} - \sum_{k \in N(i)} P_{ik} \Delta_i t_{ij} - M_{ij}(1 - x_i) \;\; \forall\, i, j \in V, i \neq j \tag{5.24f}$$

$$z_{ij} \leq M_{ij} x_i \qquad \forall\, i, j \in V, i \neq j \tag{5.24g}$$

$$z_{ij} \geq -M_{ij} x_i \qquad \forall\, i, j \in V, i \neq j \tag{5.24h}$$

$$t_{ij} \geq 0 \qquad \forall\, i, j \in V, i \neq j \tag{5.24i}$$

$$x_i \in \{0, 1\} \qquad \forall\, i \in V. \tag{5.24j}$$

As it can be seen, this formulation contains several big-M's which considering constraints (5.24e)–(5.24f), must satisfy the following inequality:

$$M_{ij} \geq \sum_{\substack{k \in N(i) \\ k \neq j}} \Delta_i P_{ik} t_{kj} - \sum_{k \in N(i)} \Delta_i P_{ik} t_{ij} \tag{5.25}$$

Therefore, M_{ij} should be the upper bound of the right-hand side of inequality (5.25) which is obtained when t_{kj} takes its upper bound and t_{ij} takes its lower bound. These bounds can be computed based on the discussion in Section 5.2:

$$M_{ij} = \sum_{\substack{k \in N(i) \\ k \neq j}} \Delta_i P_{ik} \hat{t}_{kj} - \sum_{k \in N(i)} \Delta_i P_{ik} \bar{t}_{ij}, \tag{5.26}$$

where $\hat{t}_{kj}$ is the upper bound on the value of t_{kj} obtained by interdicting all the vertices in the network and $\bar{t}_{ij}$ is the lower bound on the value of t_{ij} which is the expected first passage time from i to j in the original graph.

5.5 Computational experiments

In this section, we present the result of implementing Formulation (5.24) on the instances introduced in Table 4.1. All the graphs in this testbed are undirected, and we convert them to directed graphs. Experiments are conducted on a 64-bit Windows® 10 Pro machine with 16GB of RAM and 1.8 GHz processor with 7 cores. All algorithms are implemented in C++, compiled using Microsoft® Visual Studio® 2017, and Gurobi™ Optimizer v9.5.2 is used to solve the MILPs (Gurobi Optimization, LLC, 2021).

Before implementing Formulation (5.24), we use Gurobi to find the first passage times in the original graph for all the instances in Table 4.1. For some of the graphs in this testbed, the solver hits the one hour time limit or reports an "out-of-memory" error when solving the system of linear equations to compute the first passage times. We exclude these instances from our experiments considering that Formulation (5.24) has many variables and constraints in addition to a system of linear equations formed by constraints (5.24d).

In our experiments, sets S and T are chosen randomly and their cardinality is equal to 20% of the number of vertices. The interdiction budget B is also equal to 20% of the number of vertices. For every vertex $i \in V$, transition probabilities of its loop and outgoing arcs are computed as $1/(\deg(i)+1)$ and $\Delta_i = 0.5$. We report the results in Table 5.1. All the instances are solved to optimality in the root node and the increase in the value of the smallest first passage time is at least 23% for `football` and at most 55% for `celegansm`.

Table 5.1: Results of implementing Formulation (5.24). FPT_G and $FPT_{G'}$ are the smallest first passage time from S to T respectively in the original graph and the interdicted graph.

Graph	# nodes	# edges	FPT_G	$FPT_{G'}$	increase in FPT (%)	Solution time (s)
karate	34	78	56.02	79.56	42.03	0.15
dolphins	62	159	38.77	52.60	35.66	0.53
lesmis	77	254	56.91	81.75	43.66	0.84
polbooks	105	441	54.87	77.67	41.57	2.63
adjnoun	112	425	39.69	57.13	43.96	3.86
football	115	613	100.94	124.38	23.22	7.36
jazz	198	2742	70.85	98.22	38.62	52.96
celegansn	297	2148	231.97	331.90	43.08	205.98
celegansm	453	2025	30.50	47.48	55.66	294.50

Our numerical experiments show that the objective of maximizing the smallest first passage time in a graph can be too restrictive in some cases. As an example, with the setting in our experiments, for any vertex $i \in V$ with a degree equal to one and its neighbor j, we have $t_{ij} = 2$ which is the smallest first passage time in the graph. For these pairs of vertices, regardless of the interdiction policy, it can be shown that the value of t_{ij} will not exceed 4. Also, instead of focusing on the smallest first passage times, one might decide in order to maximize a quantity, not as "extreme" as the minimum first passage time. This motivates us to consider the problem of maximizing the q-th smallest first passage time in the next section.

5.6 Maxmizing the q-th smallest first passage time

In this section, we study the interdicting problem to maximize the q-th smallest first passage time between S and T in the graph. For any $i \in S$, let $\widetilde{t}_i$ be the smallest first passage time from i to any vertex in T, i.e., $\widetilde{t}_i = \min\{t_{ij} : j \in T\}$ for every $i \in S$. Our goal is to maximize the q-th smallest $\widetilde{t}_i$ with $q = 1, 2, ..., |S|$. That is, we seek to solve:

$$\max \left\{ \min\{\widetilde{t}_i(x) : i \in S\} : \sum_{i \in V} x_i \leq B \right\}, \tag{5.27}$$

where $\widetilde{t}_i(x)$ is the value of $\widetilde{t}_i$ after the vertices indicated by x are interdicted.

We define binary variable y_i that takes one whenever the value of variable $\widetilde{t}_i$ is one of the $(q-1)$ smallest $\widetilde{t}_i$s. Using the same variables defined for the Formulation (5.24), we have:

$$\max\ \theta \tag{5.28a}$$

$$\text{s.t.} \sum_{i \in V} y_i = q - 1 \tag{5.28b}$$

$$\widetilde{t}_i y_i \leq \theta \qquad \forall\, i \in V \tag{5.28c}$$

$$\theta \leq (M - \widetilde{t}_i) y_i + \widetilde{t}_i \qquad \forall\, i \in V \tag{5.28d}$$

$$\widetilde{t}_i \leq t_{ij} \qquad \forall\, i, j \in V, i \neq j \tag{5.28e}$$

$$\sum_{i \in V} x_i \leq B \tag{5.28f}$$

$$t_{ij} = 1 + \sum_{\substack{k \in N(i) \\ k \neq j}} P_{ik}(1 - \Delta_i x_i) t_{kj} + \Big(P_{ii} + \sum_{k \in N(i)} P_{ik} \Delta_i x_i\Big) t_{ij} \qquad \forall\, i, j \in V, i \neq j \tag{5.28g}$$

$$t_{ij} \geq 0 \qquad \forall\, i, j \in V \tag{5.28h}$$

$$x_i \in \{0, 1\} \qquad \forall\, i \in V \tag{5.28i}$$

$$y_i \in \{0, 1\} \qquad \forall\, i \in V. \tag{5.28j}$$

Constraint (5.28d) contains a big-M that must be at least the size of the q-th smallest $\widetilde{t}_i$. Also, constraints (5.28d) and (5.28g) are nonlinear. The latter is the same as the constraint (5.22d) in Formulation (5.22) and can be linearized in the same way. To linearize constraint (5.28d), we introduce a new continuous variable $u_i = \widetilde{t}_i y_i$, and we will have:

$$\max\ \theta \tag{5.29a}$$

$$\sum_{i \in V} y_i = q - 1 \tag{5.29b}$$

$$u_i \leq \theta \qquad \forall\, i \in V \tag{5.29c}$$

$$\theta \leq M y_i - u_i + \widetilde{t}_i \qquad \forall\, i \in V \tag{5.29d}$$

$$u_i \leq \widetilde{t}_i + M(1 - y_i) \qquad \forall\, i \in V \tag{5.29e}$$

$$u_i \geq \widetilde{t}_i - M(1 - y_i) \qquad \forall\, i \in V \tag{5.29f}$$

$$u_i \leq M y_i \qquad \forall\, i \in V \tag{5.29g}$$

$$\widetilde{t}_i \leq t_{ij} \qquad \forall\, i \in V, j \in V, i \neq j \tag{5.29h}$$

$$\sum_{i \in V} x_i \leq B \tag{5.29i}$$

$$t_{ij} = 1 - z_{ij} + \sum_{\substack{k \in N(i) \\ k \neq j}} P_{ik} t_{kj} + P_{ii} t_{ij} \qquad \forall\, i, j \in V, i \neq j \tag{5.29j}$$

$$z_{ij} \leq \sum_{\substack{k \in N(i) \\ k \neq j}} P_{ik} \Delta_i t_{kj} - \sum_{k \in N(i)} P_{ik} \Delta_i t_{ij} + M_{ij}(1 - x_i) \qquad \forall\, i, j \in V, i \neq j \tag{5.29k}$$

$$z_{ij} \geq \sum_{\substack{k \in N(i) \\ k \neq j}} P_{ik} \Delta_i t_{kj} - \sum_{k \in N(i)} P_{ik} \Delta_i t_{ij} - M_{ij}(1 - x_i) \qquad \forall\, i, j \in V, i \neq j \tag{5.29l}$$

$$z_{ij} \leq M_{ij} x_i \qquad \forall\, i, j \in V, i \neq j \tag{5.29m}$$

$$z_{ij} \geq -M_{ij} x_i \qquad \forall\, i, j \in V, i \neq j \tag{5.29n}$$

$$t_{ij} \geq 0 \qquad \forall\, i, j \in V \tag{5.29o}$$

$$u_{ij} \geq 0 \qquad \forall\, i, j \in V \tag{5.29p}$$

$$x_i \in \{0, 1\} \qquad \forall\, i \in V \tag{5.29q}$$

$$y_i \in \{0, 1\} \qquad \forall\, i \in V. \tag{5.29r}$$

Next, we show the correctness of Formulation (5.28). Consider graph $G = (V, E)$. Let $D \subseteq V$ be the interdiction set and x^D be its characteristic vector. Theorem (6) proves the correctness of Formulation (5.28).

Theorem 6. *A subset $D \subseteq V$ is a maximizer for problem* (5.27) *if and only if there exist $x^D, y^D, t^D, \widetilde{t}^D$ such that $(\theta^*, x^D, y^D, t^D, \widetilde{t}^D)$ is an optimal solution to Formulation* (5.28).

($\Longrightarrow$) Suppose D is a maximizer for the original problem. We want to show that there exist $x^D, y^D, t^D, \widetilde{t}^D$ such that $(\theta^*, x^D, y^D, t^D, \widetilde{t}^D)$ is an optimal solution to Formulation (5.28) with the objective value equal to $c^* = \theta^*$.

For the sake of contradiction, suppose there is no $x^D, y^D, t^D, \widetilde{t}^D$ such that $(\theta^*, x^D, y^D, t^D, \widetilde{t}^D)$ is an optimal solution to Formulation (5.28). Thus, because it can be readily checked that an optimal solution to Formulation (5.28) always exists, we can assume that there exists a solution $(\bar{\theta}, x^{\bar{D}}, y^{\bar{D}}, t^{\bar{D}}, \widetilde{t}^{\bar{D}})$ that is optimal with the objective value $\bar{c} = \bar{\theta}$ such that $\bar{D} \neq D$ and $\bar{c} > c^*$.

Based on the constraints (5.28c) and (5.28d), if $y_i^{\bar{D}} = 0$, then $0 \leq \bar{\theta} \leq \widetilde{t}_i^{\bar{D}}$, and if $y_i^{\bar{D}} = 1$, then $\widetilde{t}_i^{\bar{D}} \leq \bar{\theta} \leq M$. On the other hand, based on the constraint (5.28b), only $(q-1)$ of variables y_i can take one and others will take zero. Considering these three types of constraints, the solution $(\bar{\theta}, x^{\bar{D}}, y^{\bar{D}}, t^{\bar{D}}, \widetilde{t}^{\bar{D}})$ is feasible if $y_i^{\bar{D}} = 1$ for those $i \in V$ such that $\widetilde{t}_i^{\bar{D}}$ is in the $(q-1)$ smallest values, otherwise, the lower bounds in constraints (5.28c) can become larger than the upper bounds in constraints (5.28d) which results in the infeasibility of the solution.

Since Formulation (5.28) maximizes the objective function, and variable θ has a positive objective function coefficient, it takes the largest possible value considering the bounds in constraints (5.28c) and (5.28d). So, $\bar{\theta}$, the value of variable θ given $\bar{D}$, will be exactly equal to the smallest value in the RHS of constraints (5.28d), i.e., $\min \big\{M, \min\{\widetilde{t}_i^{\bar{D}} | i \in V, y_i = 0\}\big\}$. As we mentioned above, we have $y_i^{\bar{D}} = 1$ for $i \in V$ associated with $(q-1)$ smallest $\widetilde{t}_i^{\bar{D}}$ and $y_i^{\bar{D}} = 0$ for $i \in V$ associated with $(n-q+1)$ largest $\widetilde{t}_i^{\bar{D}}$. This means that for $i \in V$ associated with the q-th smallest $\widetilde{t}_i^{\bar{D}}$, we have $y_i^{\bar{D}} = 0$. Therefore, $\bar{\theta} = \bar{c} = \{M, \min\{\widetilde{t}_i^{\bar{D}} | i \in V, y_i^{\bar{D}} = 0\}\}$ will be equal the q-th smallest $\widetilde{t}_i^{\bar{D}}$ which is equivalent to $\min\{\widetilde{t}_i(x) : i \in S\}$ according to the original problem.

This shows that D is not a maximizer for the original problem because we assumed $\bar{c} > c^*$ and showed that $\bar{c} = \min\{\widetilde{t}_i(x^{\bar{D}}) : i \in S\}$, which means that the q-th smallest $\widetilde{t}_i$ in the graph by interdicting set $\bar{D}$ is greater than the q-th smallest $\widetilde{t}_i$ by interdicting D. This is a contradiction and we can conclude that $(\theta^*, x^D, y^D, t^D, \widetilde{t}^D)$ is an optimal solution to Formulation (5.28).

($\Longleftarrow$) Suppose $(\theta^*, x^D, y^D, t^D, \widetilde{t}^D)$ with the objective value equal to $c^* = \theta^*$ is an optimal solution to Formulation (5.28). We want to show that D is a maximizer for problem (5.27).

For the sake of contradiction, suppose D is not a maximizer and the original problem has a maximizer $\bar{D}$ where $\bar{D} \neq D$ such that $\min\{\widetilde{t}_i(x^{\bar{D}}) : i \in S\} > \min\{\widetilde{t}_i(x^D) : i \in S\}$. We show that if this holds, then $(\theta^*, x^D, y^D, t^D, \widetilde{t}^D)$ cannot be an optimal solution to Formulation (5.28). Based on the discussion in the other direction of the proof, we know that variable θ in Formulation (5.27) will be equal to the smallest RHS of constraints (5.28d). We also showed that the RHS of this constraint is equal to M for $(q-1)$ of the constraints associated with those $i \in V$ where $\widetilde{t}_i$s have smaller values and is equal to $\widetilde{t}_i$ for $(n-q+1)$ of the constraints associated with those $i \in V$ where $\widetilde{t}_i$s have larger values. Therefore, $\bar{\theta}$, value of variable θ given $\bar{D}$, will be equal to the q-th smallest $\widetilde{t}_i$, i.e., $\min\{\widetilde{t}_i(x^{\bar{D}}) : i \in S\}$.

Since we assumed $\min\{\widetilde{t}_i(x^{\bar{D}}, y^{\bar{D}}) : i \in S\} > \min\{\widetilde{t}_i(x^D, y^D) : i \in S\}$, we can conclude that $\bar{\theta} > \theta^*$. This shows that solution $(\theta^*, x^D, y^D, t^D, \widetilde{t}^D)$ is not optimal. This is a contradiction and we can conclude that D is a maximizer for the original problem. This proves our desired result, so Formulation (5.28) is correct.$\square$

Finally, we present the results of implementing Formulation (5.28) on the same instances used in Section 5.5. Our setting is also the same as the previous experiments with probabilities to be $\frac{1}{\deg(i)+1}$, 20% of vertices as the budget and randomly chosen sets S and T. We set

q to be half of the number of vertices in set S meaning that the interdictor ignores the ($|S|/2$) smallest first passage times and focuses on maximizing the other ($|S|/2$) largest first passage times. The results are presented in Table 5.2. Solving Formulation (5.28) has been more difficult than Formulation (5.24) due to more variables and constraints. It can be seen that the running times have increased for all the instances and `celegansm` is not solved to optimality in one hour time limit. However, the increase in the value of the first passage times is between 22% to 50%. For both formulations (5.24) and (5.28), the direct implementation does not allow us to solve the problem for large instances. We will discuss the possible improvements in Chapter VI.

Table 5.2: Results of implementation of Formulation (5.29) with $q = |V|/10$.

Graph	FPT_G	$FPT_{G'}$	increase in FPT (%)	#B&B nodes	Solution time (s)	Gap (%)
karate	63.36	90.19	42.35	1	0.63	0.00
dolphins	52.59	71.37	35.71	1	2.70	0.00
lesmis	72.47	103.32	42.58	1	3.81	0.00
polbooks	118.49	175.81	48.38	1	11.77	0.00
adjnoun	51.23	73.26	43.01	1	15.64	0.00
football	126.46	154.51	22.18	169	78.45	0.00
jazz	79.31	109.58	38.17	221	704.34	0.00
celegansn	239.17	342.07	43.02	2300	2645.06	0.00
celegansm	49.52	74.54	50.52	1133	3620.53	8.42

5.7 Comparing interdiction policies

In this section, we compare the effect of different interdiction policies on the increase of the value of the first passage times. We consider the interdiction policies generated by four approaches: 2-club interdiction and 3-club interdiction in Chapter IV, first passage time interdiction introduced in this chapter, and a random interdiction policy where 20% of vertices are selected randomly to be interdicted. We compute the value of the smallest first passage times in the interdicted graph using the interdiction set generated by each method. For s-club interdiction approaches, we have considered the solutions when $\alpha = 2$ (Results in Tables 4.3 and 4.6). All the other parameters are the same as the experiments in Section 5.5.

Table 5.3 shows the results of these experiments. Column FPT_G shows the value of the smallest first passage time in the original graph and columns $FPT_{G'}$ show the smallest first passage time after interdiction. NA values indicate that we were not able to generate an interdiction policy by that method to use in these experiments. It can be seen that by interdicting the same number of vertices, the increase obtained by the first passage time interdiction is always larger than the random interdiction which shows the usefulness of this approach.

FPT interdiction is outperforming the 2-club and 3-club interdiction methods for all the instances. This observation is in line with our expectations because unlike s-club interdiction methods that find the policies to decrease the size of the cohesive subgroups, the objective of the FPT interdiction is to increase the first passage times. In addition, we should note that the s-club interdiction methods use an interdiction penalty instead of a budget, and as a result, the size of the interdiction sets obtained by these methods might be smaller. For example, for `dolphins` and `football`, no vertex is interdicted by the 2-club interdiction approach. This difference is another explanation for the smaller increase in the first passage times in comparison to the FPT interdiction or even the random interdiction approach considering Proposition 6 in Section 5.2.

Table 5.3: Comparing the increase in the value of the smallest first passage time using different interdiction policies.

		FPT Interdiction		2-club interdiction		3-club interdiction		Random interdiction	
Graph	FPT_G	$FPT_{G'}$	Increase (%)	$FPT_{G'}$	Increase (%)	$FPT_{G'}$	Increase (%)	$FPT_{G'}$	Increase (%)
karate	56.02	79.56	42.03	70.42	25.72	77.12	37.68	67.69	20.85
dolphins	38.77	52.60	35.66	38.77	0.00	42.15	8.70	45.71	17.89
lesmis	56.91	81.75	43.66	62.60	10.02	71.85	26.25	62.74	10.26
polbooks	54.87	77.67	41.57	56.48	2.94	67.25	22.57	63.39	15.54
adjnoun	39.69	57.13	43.96	45.28	14.08	48.50	22.21	41.70	5.07
football	100.94	124.38	23.22	100.94	0.00	102.75	1.79	121.38	20.25
jazz	70.85	98.22	38.62	NA	NA	NA	NA	83.10	17.28
celegansn	231.97	331.90	43.08	270.54	16.62	NA	NA	279.56	20.51
celegansm	30.50	47.48	55.66	37.62	23.35	39.48	29.44	38.18	25.10

CHAPTER VI

CONCLUSION AND FUTURE WORK

In this dissertation, we studied interdicting cohesive subgroups and interdicting first passage times in networks. We have focused on the application of these methodologies in online social network analysis and presented formulations and algorithms to solve these problems on the benchmark instances. In this section, we summarize our contributions and identify possible future research directions.

6.1 Contributions

The first problem we studied in this dissertation is the maximum s-club interdiction. This study makes the following contributions to the literature of adversarial community disruptions, specifically, interdiction by deletion of vertices in maximum cardinality s-clubs.

We introduce the new concept of *H-hereditary s-clubs*, which extends the notion of heredity to s-clubs. Based on H-heredity, we introduce an MILP formulation of the s-club interdiction problem that has fewer constraints than the naive MILP formulation that is based on standard interdiction formulation techniques. We show that the LP relaxation of the proposed formulation does not have redundant constraints. We also derive three types of facet defining inequalities for the convex hull of feasible solutions by further strengthening the new constraints based on H-heredity for special s-clubs.

We establish a one-to-one correspondence between the sets inducing H-heredity in an s-club and latency-s connected dominating sets (latency-s CDSs) of the s-club (Validi and

Buchanan, 2020). We exploit this relationship in a decomposition branch-and-cut algorithm based on delayed constraint generation. This approach is able to solve several real-life and synthetic instances of the interdiction problem with more than 10,000 vertices in a matter of minutes. Moreover, our approach solves the problem orders of magnitude faster than using an analogous algorithm based on the naive MILP formulation.

The second problem is the first passage time interdiction. We used the first passage times to measure how fast misinformation is disseminated in an online network, and interdict vertices to delay the propagation times. This is a new approach in the literature of online social network analysis and network interdiction with non-deterministic evaders. We considered two different objectives; maximizing the smallest first passage time and maximizing the q-th smallest first passage time between two given sets of accounts. For each problem, we have proposed MILP formulations and solved the problem on the benchmark instances. We also show the problem is NP-hard by reduction from the vertex cover problem.

6.2 Future work

For the maximum s-club interdiction problem, we used two different groups of instances and our numerical results show that the 3-club interdiction problem is still quite challenging to solve on the second group of instances in our test-bed on which solving the NP-hard maximum 3-club problem remains difficult. Given the importance of conclusive termination during separation calls for the correctness of such a relaxation-based decomposition branch-and-cut scheme, further breakthroughs are needed to solve the maximum s-club and minimum latency-s CDS problems on this test bed for $s \geq 3$. The initial relaxation also needs further investigation and strengthening, especially for $s \geq 3$, to shift the computational burden away from the separation procedures to the extent possible. These developments and improved inexact separation procedures for $s \geq 3$ can further extend our ability to solve the s-club interdiction on even larger scale social networks.

For the first passage time interdiction problem, we will study the possibility of proposing polynomial time algorithms to solve the problem on special graphs such as paths and trees. In our experiments, we have implemented the MILP formulations directly and as a result, we are not able to solve large instances due to the number of variables and constraints in the formulations. It is beneficial to investigate techniques to design decomposition algorithms for these formulations. We will also improve the formulations by finding bounds on the value of the first passage times using the structure of the network and we will consider using block decomposition algorithms to find vertices whose deletion can result in disconnecting the graph and increasing the first passage times values.

Additionally, considering the existing studies on the misinformation blocking problem, and the network interdiction approaches introduced in this dissertation, it would be beneficial to conduct a simulation study to compare the effectiveness of these methodologies in minimizing the spread of misinformation in real-world social networks under different models of propagation. Such studies would be similar to our comparisons in Section 5.7, with the difference that they would be based on discrete-event simulations of other models of propagation such as e.g., the independent cascade model or the linear threshold model of Kempe et al. (2003). We anticipate that using our proposed interdiction policies would result in a substantial decrease in the spread of harmful cascades, when compared to easy-to-implement benchmark policies such as, e.g., deleting vertices of the network at random.

REFERENCES

R. D. Alba. A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3(1):113–126, 1973.

H. Allcott and M. Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36, 2017.

N. Assimakopoulos. A network interdiction model for hospital infection control. *Computers in biology and medicine*, 17(6):413–422, 1987.

B. Balasundaram and F. M. Pajouh. Graph theoretic clique relaxations and applications. In P. M. Pardalos, D.-Z. Du, and R. Graham, editors, *Handbook of Combinatorial Optimization*, pages 1559–1598. Springer, New York, 2nd edition, 2013. ISBN 978-1-4419-7996-4.

B. Balasundaram, S. Butenko, and S. Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10(1):23–39, August 2005.

B. Balasundaram, S. Butenko, and I. V. Hicks. Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*, 59(1):133–142, 2011.

M. O. Ball, B. L. Golden, and R. V. Vohra. Finding the most vital arcs in a network. *Operations Research Letters*, 8(2):73–76, 1989.

A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439): 509–512, 1999.

J. Berkhout and B. F. Heidergott. Analysis of markov influence graphs. *Operations Research*, 67(3):892–904, 2019.

J. S. Borrero, O. A. Prokopyev, and D. Sauré. Sequential shortest path interdiction with incomplete information. *Decision Analysis*, 13(1):68–98, 2016.

J. S. Borrero, O. A. Prokopyev, and D. Sauré. Sequential interdiction with incomplete information and learning. *Operations Research*, 67(1):72–89, 2019.

J. S. Borrero, O. A. Prokopyev, and D. Sauré. Learning in sequential bilevel linear programming. *INFORMS Journal on Optimization*, 4(2):174–199, 2022.

J.-M. Bourjolly, G. Laporte, and G. Pesant. Heuristics for finding k-clubs in an undirected graph. *Computers & Operations Research*, 27:559–569, 2000.

J.-M. Bourjolly, G. Laporte, and G. Pesant. An exact algorithm for the maximum k-club problem in an undirected graph. *European Journal Of Operational Research*, 138:21–28, 2002.

C. Brezovec, G. Cornuéjols, and F. Glover. Two algorithms for weighted matroid intersection. *Mathematical Programming*, 36(1):39–53, 1986.

S. Butenko and W. Wilhelm. Clique-detection models in computational biochemistry and genomics. *European Journal of Operational Research*, 173:1–17, 2006.

M. Cerulli, D. Serra, C. Sorgente, C. Archetti, and I. Ljubic. Mathematical programming formulations for the collapsed k-core problem. *arXiv preprint arXiv:2211.14833*, 2022.

F. Chung and L. Lu. *Complex Graphs and Networks*. CBMS Lecture Series. American Mathematical Society, Providence, RI, 2006.

M. Cochet. Minors and social media- how are the most vulnerable protected? https://computationalsocialmedia.tech/index.php/2021/03/02/minors-and-social-media-how-are-the-most-vulnerable-protected/, Accessed November 3, 2022, March 2021.

H. W. Corley and Y. S. David. Most vital links and nodes in weighted networks. *Operations Research Letters*, 1(4):157–160, 1982.

N. Daemi, J. S. Borrero, and B. Balasundaram. Decomposition branch-and-cut solver for 3-club interdiction. C++ Codes online at: https://github.com/niloufardaemi/3club_interdiction, Accessed January 6, 2022, August 2021a.

N. Daemi, J. S. Borrero, and B. Balasundaram. Decomposition branch-and-cut solver for 2-club interdiction. C++ Codes online at: https://github.com/niloufardaemi/2club_interdiction, Accessed January 6, 2022, August 2021b.

N. Daemi, J. S. Borrero, and B. Balasundaram. Interdicting low-diameter cohesive subgroups in large-scale social networks. *INFORMS Journal on Optimization*, 4(3):304–325, 2022.

DIMACS. Graph Partitioning and Graph Clustering: Tenth DIMACS Implementation Challenge. http://www.cc.gatech.edu/dimacs10/index.shtml, 2012. Accessed Feb 2015.

E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

P. Domm. False rumor of explosion at white house causes stocks to briefly plunge; ap confirms its twitter feed was hacked. *CNBC. COM*, 23:2062, 2013.

U. Feige, V. S. Mirrokni, and J. Vondrák. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

M. Fischetti, M. Monaci, and M. Sinnl. A dynamic reformulation heuristic for generalized interdiction problems. *European Journal of Operational Research*, 267(1):40–51, 2018.

M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. Interdiction games and monotonicity, with application to knapsack problems. *INFORMS Journal on Computing*, 31(2):390–410, 2019.

F. Furini, I. Ljubić, S. Martin, and P. San Segundo. The maximum clique interdiction problem. *European Journal of Operational Research*, 277(1):112–127, 2019.

N. Gould and J. Scott. A note on performance profiles for benchmarking software. *ACM Transactions on Mathematical Software (TOMS)*, 43(2):1–5, 2016.

M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd international conference on World Wide Web*, pages 729–736, 2013.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2021. URL `http://www.gurobi.com`.

A. Gutfraind, A. Hagberg, and F. Pan. Optimal interdiction of unreactive markovian evaders. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 102–116. Springer, 2009.

D. L. Han, L. C. Tang, and H. C. Huang. A markov model for single-leg air cargo revenue management under a bid-price policy. *European Journal of Operational Research*, 200(3):800–811, 2010.

Help Net Security. Attackers use large-scale bots to launch attacks on social media platforms. https://www.helpnetsecurity.com/2019/08/27/attacks-on-social-media-platforms, Accessed October 31, 2022, August 2019.

E. Israeli and R. K. Wood. Shortest-path network interdiction. *Networks: An International Journal*, 40(2):97–111, 2002.

M. P. Johnson, A. Gutfraind, and K. Ahmadizadeh. Evader interdiction: algorithms, complexity and collateral damage. *Annals of operations research*, 222(1):341–359, 2014.

D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

V. G. Kulkarni. *Modeling and analysis of stochastic systems*, chapter 3. Chapman and Hall/CRC, 2016.

J. Kunegis. Konect: the Koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1343–1350, 2013.

J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data/, June 2014.

O. Lesser, L. Tenenboim-Chekina, L. Rokach, and Y. Elovici. Intruder or welcome friend: Inferring group membership in online social networks. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 368–376. Springer, 2013.

J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is np-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

Y. Lu, E. Moradi, and B. Balasundaram. Correction to: Finding a maximum k-club using the k-clique formulation and canonical hypercube cuts. *Optimization Letters*, 12(8):1959–1969, November 2018.

R. D. Luce and A. D. Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116, 1949.

J. Luo, H. Molter, and O. Suchỳ. A parameterized complexity view on collapsing k-cores. *Theory of Computing Systems*, 65(8):1243–1282, 2021.

K. Malik, A. K. Mittal, and S. K. Gupta. The k most vital arcs in the shortest path problem. *Operations Research Letters*, 8(4):223–227, 1989.

D. W. Matula and L. L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM (JACM)*, 30(3):417–427, 1983.

M. I. Meltzer, I. Damon, J. W. LeDuc, and J. D. Millar. Modeling potential responses to smallpox as a bioterrorist weapon. *Emerging Infectious Diseases*, 7(6):959, 2001.

R. J. Mokken. Cliques, clubs and clans. *Quality and Quantity*, 13(2):161–173, 1979.

J. W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, 1965.

E. Moradi and B. Balasundaram. Finding a maximum k-club using the k-clique formulation and canonical hypercube cuts. *Optimization Letters*, 12(8):1947–1957, November 2018.

D. P. Morton, F. Pan, and K. J. Saeger. Models for nuclear smuggling interdiction. *IIE Transactions*, 39(1):3–14, 2007.

A. K. Nandi and H. R. Medal. Methods for removing links in a network to minimize the spread of infections. *Computers & Operations Research*, 69:10–24, 2016.

M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

F. M. Pajouh and B. Balasundaram. On inclusionwise maximal and maximum cardinality k-clubs in graphs. *Discrete Optimization*, 9(2):84–97, May 2012.

F. Pan, W. Charlton, and D. Morton. Interdicting smuggled nuclear material. In D. Woodruff, editor, *Network Interdiction and Stochastic Integer Programming*, pages 1–20. Kluwer Academic Publishers, Boston, 2003.

S. Pasupuleti. Detection of protein complexes in protein interaction networks using n-clubs. In *In EvoBIO 2008: Proceedings of the 6th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pages 153–164. Springer, 2008. volume 4973 of Lecture Notes in Computer Science.

J. Pattillo, N. Youssef, and S. Butenko. On clique relaxation models in network analysis. *European Journal of Operational Research*, 226(1):9–18, 2013.

C. V. Pham, Q. V. Phu, H. X. Hoang, J. Pei, and M. T. Thai. Minimum budget for misinformation blocking in online social networks. *Journal of Combinatorial Optimization*, 38(4):1101–1127, 2019.

S. Raghavan and R. Zhang. A branch-and-cut approach for the weighted target set selection problem on social networks. *INFORMS Journal on Optimization*, 1(4):304–322, 2019.

K. Rapoza. Can "fake news" impact the stock market? `https://www.forbes.com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/?sh=7cae6dcb2fac`, Accessed November 3, 2022, February 2017.

R. A. Rossi and N. K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL `http://networkrepository.com`.

H. Salemi and A. Buchanan. Parsimonious formulations for low-diameter clusters. *Mathematical Programming Computation*, 12(3):493–528, 2020.

J. A. Sefair, J. C. Smith, M. A. Acevedo, and R. J. Fletcher Jr. A defender-attacker model and algorithm for maximizing weighted expected hitting time with application to conservation planning. *IISE Transactions*, 49(12):1112–1128, 2017.

S. B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269–287, 1983.

D. Shah and T. Zaman. Finding rumor sources on random trees. *Operations Research*, 64(3): 736–755, 2016.

P. Shi, Z. Zhang, and K.-K. R. Choo. Detecting malicious social bots based on clickstream sequences. *IEEE Access*, 7:28855–28862, 2019.

C. Silverman. This analysis shows how viral fake election news stories outperformed real news on Facebook. `https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook`, Accessed November 3, 2022, November 2016.

J. C. Smith and Y. Song. A survey of network interdiction models and algorithms. *European Journal of Operational Research*, 283(3):797–811, 2020.

T. Spangler. Twitter stock slides on report that it has been deleting over 1 million fake accounts daily. `https://variety.com/2018/digital/news/twitter-stock-deleted-fake-accounts-1202868405/`, Accessed November 3, 2022, July 2018.

K. M. Sullivan, D. P. Morton, F. Pan, and J. Cole Smith. Securing a border under asymmetric information. *Naval Research Logistics (NRL)*, 61(2):91–100, 2014.

Y. Tang, J.-P. P. Richard, and J. C. Smith. A class of algorithms for mixed-integer bilevel min–max optimization. *Journal of Global Optimization*, 66(2):225–262, 2016.

K. Tanınmış, N. Aras, and I. Altınel. Influence maximization with deactivation in social networks. *European Journal of Operational Research*, 278(1):105–119, 2019.

K. Tanınmış, N. Aras, İ. K. Altınel, and E. Güney. Minimizing the misinformation spread in social networks. *IISE Transactions*, 52(8):850–863, 2020.

K. Tanınmış, N. Aras, and İ. K. Altınel. Improved x-space algorithm for min-max bilevel problems with an application to misinformation spread in social networks. *European Journal of Operational Research*, 297(1):40–52, 2022.

H. Validi and A. Buchanan. The optimal design of low-latency virtual backbones. *INFORMS Journal on Computing*, 32(4):952–967, 2020.

J. L. Walteros and A. Buchanan. Why is maximum clique often easy in practice? *Operations Research*, 68(6):1866–1895, 2020.

S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

D. J. Welsh. *Matroid Theory*. Courier Corporation, 2010.

R. K. Wood. Deterministic network interdiction. *Mathematical and Computer Modelling*, 17 (2):1–18, 1993.

Youth Equipped To Succeed. How social media affects teens. `https://justsayyes.org/jsy-blog/how-social-media-affects-teens/`, Accessed November 3, 2022, October 2022.

F. Zhang, Y. Zhang, L. Qin, W. Zhang, and X. Lin. Finding critical users for social network engagement: The collapsed k-core problem. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

VITA

Niloufar Daemi

Candidate for the Degree of

Doctor of Philosophy

Dissertation: NETWORK INTERDICTION APPROACHES FOR DIMINISHING MISINFORMATION SPREAD IN SOCIAL NETWORKS

Major Field: Industrial Engineering and Management

Biographical:

Education:

Completed the requirements for the Doctor of Philosophy in Industrial Engineering and Management at Oklahoma State University, Stillwater, Oklahoma in December, 2022.

Completed the requirements for the Master of Science in Industrial Engineering at Amirkabir University of Technology, Tehran, Iran in 2016.

Completed the requirements for the Bachelor of Science in Industrial Engineering at Noshirvani University of Technology, Babol, Iran in 2012.